

# Package ‘NanoStringDiff’

April 16, 2024

**Type** Package

**Title** Differential Expression Analysis of NanoString nCounter Data

**Version** 1.32.0

**Date** 2018-01-24

**Imports** matrixStats, methods, Rcpp

**LinkingTo** Rcpp

**Depends** Biobase

**Suggests** testthat, BiocStyle

**Author** hong wang <hong.wang@uky.edu>, tingting zhai <tingting.zhai@uky.edu>, chi wang <chi.wang@uky.edu>

**Maintainer** tingting zhai <tingting.zhai@uky.edu>, hong wang <hong.wang@uky.edu>

**Description** This Package utilizes a generalized linear model (GLM) of the negative binomial family to characterize count data and allows for multi-factor design. NanoStringDiff incorporates size factors, calculated from positive controls and housekeeping controls, and background level, obtained from negative controls, in the model framework so that all the normalization information provided by NanoString nCounter Analyzer is fully utilized.

**License** GPL

**biocViews** DifferentialExpression, Normalization

**NeedsCompilation** yes

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

**git\_branch** RELEASE\_3\_18

**git\_last\_commit** 4afdd7f

**git\_last\_commit\_date** 2023-10-24

**Repository** Bioconductor 3.18

**Date/Publication** 2024-04-15

**R topics documented:**

NanoStringDiff-package . . . . .	2
estNormalizationFactors . . . . .	3
glm.LRT . . . . .	4
housekeepingControl . . . . .	5
housekeepingFactor . . . . .	6
NanoStringData . . . . .	7
NanoStringDataNormalization . . . . .	8
NanoStringSet-class . . . . .	8
negativeControl . . . . .	10
negativeFactor . . . . .	11
PlotsPositiveHousekeeping . . . . .	12
positiveControl . . . . .	13
positiveFactor . . . . .	14
<b>Index</b>	<b>16</b>

---

NanoStringDiff-package

*A short title line describing what the package does*

---

**Description**

A more detailed description of what the package does. A length of about one to five lines is recommended.

**Details**

This section should provide a more detailed overview of how to use the package, including the most important functions.

**Author(s)**

Your Name, email optional.

Maintainer: Your Name <your@email.com>

**References**

This optional section can contain literature or other references for background information.

**See Also**

Optional links to other man pages

## Examples

```
## Not run:  
## Optional simple examples of the most important functions  
## These can be in \dontrun{} and \donttest{} blocks.  
  
## End(Not run)
```

---

estNormalizationFactors

*estimate normalization factors, include positive size factors, background noise, housekeeping size factors.*

---

## Description

This function estimates positive size factors, background noise and housekeeping size factors for the input "NanoStringSet" object and return the same object with positiveFactor, negativeFactor and housekeepingFactor slots filled or replaced.

## Usage

```
estNormalizationFactors(NanoStringData)
```

## Arguments

NanoStringData An object of "NanoStringSet" class.

## Value

The same "NanoStringSet" object with positiveFactor, negativeFactor and housekeepingFactor field filled or replaced.

## Author(s)

hong wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

## Examples

```
data(NanoStringData)  
NanoStringData=estNormalizationFactors(NanoStringData)  
pf=positiveFactor(NanoStringData)  
nf=negativeFactor(NanoStringData)  
hf=housekeepingFactor(NanoStringData)
```

glm.LRT

*perform gene-wise likelihood ratio test for NanoString Data***Description**

The method considers a generalized linear model of the negative binomial family to characterize count data and allows for multi-factor design. The method propose an empirical Bayes shrinkage approach to estimate the dispersion parameter and use likelihood ratio test to obtain p-value.

**Usage**

```
glm.LRT(NanoStringData,design.full,Beta=ncol(design.full), contrast=NULL)
```

**Arguments**

NanoStringData	An object of "NanoStringSet" class.
design.full	numeric matrix giving the design matrix for the generalized linear models under full model. must be of full column rank.
Beta	integer or character vector indicating which coefficients of the linear model are to be tested equal to zero. Values must be columns or column names of design. Defaults to the last coefficient. Ignored if contrast is specified.
contrast	numeric vector or matrix specifying one or more contrasts of the linear model coefficients to be tested equal to zero.

**Value**

A list	
table	A data frame with each row corresponding to a gene. Rows are sorted according to likelihood ratio test statistics. The columns are: logFC: log fold change between two groups. lr: likelihood ratio test statistics. pvalue: p-value. qvalue: adjust p-value using the procedure of Benjamini and Hochberg.
dispersion	a vector of dispersion
log.dispersion	a vector of log dispersion: log.dispersion=log(dispersion)
design.full	numeric matrix giving the design matrix under full generalizedlinear model.
design.reduce	numeric matrix giving the design matrix under reduced generalizedlinear model.
Beta.full	coefficients under full model.
mean.full	mean value under full model.
Beta.reduce	coefficients under reduced model.
mean.reduce	mean value under reduced model.
m0	hyper-parameter: mean value of the prior distribution of log dispersion
sigma	hyper-parameter: standard deviation of the prior distribution of log dispersion

**Author(s)**

hong wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**Examples**

```
data(NanoStringData)
NanoStringData=estNormalizationFactors(NanoStringData)
group=pData(NanoStringData)
design.full=model.matrix(~0+factor(group$group))
contrast=c(1,-1)
result=glm.LRT(NanoStringData,design.full,
               Beta=ncol(design.full),contrast=contrast)
head(result$table)
```

---

housekeepingControl    *Accessor functions for the 'housekeepingControl' slot in a NanoStringSet object.*

---

**Description**

user-defined housekeeping control genes can be used to estimate housekeeping factors to adjust variation caused by different sample input.

**Usage**

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

**Arguments**

object            A NanoStringSet object.  
value            A matrix with housekeeping control genes.

**Details**

NanoString nCounter analyzer also contains probes for a set of species-specific mRNA housekeeping(reference) genes that are not spike-in the system. Nanostring recommends at least three housekeeping genes, but the more that are included, the more accurate the normalization will be. Housekeeping control genes are expected consistent in their expression levels.

**Value**

A matrix contain housekeeping control genes

**Author(s)**

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**See Also**

housekeepingFactor

**Examples**

```
data(NanoStringData)
## obtain housekeeping control genes
housekeepingControl(NanoStringData)

## assign a matrix
n=ncol(exprs(NanoStringData))
r=nrow(housekeepingControl(NanoStringData))
housekeeping=matrix(rpois(r*n,1000),ncol=n)
housekeepingControl(NanoStringData)=housekeeping
```

---

housekeepingFactor	<i>Accessor functions for the 'housekeepingFactor' slot in a NanoStringSet object.</i>
--------------------	--

---

**Description**

Housekeeping size factors can be used to adjust the variance caused by different sample input.

**Usage**

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

**Arguments**

object	A NanoStringSet object.
value	A vector of housekeeping size factors.

**Details**

Housekeeping gene normalization corrects for different in sample input between assays, since reference genes are supposed to have the same expression rate between samples. So the read counts from housekeeping genes, after subtracting background noise and adjusting by positive size factors, that are not expected to vary between samples. If there exist differences, which should be caused by sample input variation.

**Value**

A vector contain housekeeping factors

**Author(s)**

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**See Also**

housekeepingControl

**Examples**

```
data(NanoStringData)
## obtain housekeeping factors
housekeepingFactor(NanoStringData)

## assign a vector
n=ncol(exprs(NanoStringData))
housekeepingFactor(NanoStringData)=rep(1,n)
```

---

NanoStringData	<i>A real 'NanoStringSet' object.</i>
----------------	---------------------------------------

---

**Description**

The object is created based on Mori Data with normal and tumor groups and 2 samples in each group. The object contain 599 endogenes, 6 positive control, 6 negative control and 4 housekeeping control.

**Usage**

```
data(NanoStringData)
```

**Value**

An object of NanoStringSet

**Examples**

```
data(NanoStringData)
NanoStringData
```

---

 NanoStringDataNormalization

*Normalize NanoStringData*


---

### Description

This function is used to get Normalized NanoString Data after adjusting for positive size factors, background noise and housekeeping size factors. Note that the normalized data values should only be used for data exploration / visualization purposes, e.g. drawing a heatmap. To perform differential expression analysis, we recommend users to follow the procedure described in the package vignette.

### Usage

```
NanoStringDataNormalization(path=path, header=TRUE, designs)
```

### Arguments

path	the path of the file which the data are to be read from.
header	a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.
designs	a data frame in which the length of vector matches the column number of NanoStringData

### Author(s)

hong wang <hong.wang@uky.edu> tingting zhai <tingting.zhai@uky.edu> chi wang <chi.wang@uky.edu>

### Examples

```
##path="/Users/NanoStringdiff-Rcode/Data/horbinski.csv"
##designs=data.frame(control=c(0,0,0,1,1,1))
##NanoStringDataNormalization(path=path, header=TRUE, designs)
```

---

 NanoStringSet-class    *NanoStringSet object and constructors*


---

### Description

The NanoStringSet is a `s4` class used to store data from NanoString nCounter analyzer. This class a subclass of `ExpressionSet`, with six more slots: `positiveControl`, `negativeControl`, `housekeepingControl`, `positiveFactor`, `negativeFactor` and `housekeepingFactor`. The constructor functions `createNanoStringSet` and `createNanoStringSetFromCsv` create a `NanoStringSet` object from two types of input: separate matrix or csv files. See the vignette for examples of construction from these two input types.



**Usage**

```
createNanoStringSet(endogenous,positiveControl,negativeControl,
                    housekeepingControl,designs)
```

```
createNanoStringSetFromCsv(path, header=TRUE, designs)
```

**Arguments**

**endogenous** for matrix input: a matrix of non-negative integers of endogenes

**positiveControl** for matrix input: a matrix of non-negative integers of positive control genes. There must have 6 positive control genes order by concentrations form high to low

**negativeControl** for matrix input: a matrix of non-negative integers of negative control genes

**housekeepingControl** for matrix input: a matrix of non-negative integers of housekeeping control genes

**designs** for data.frame input: phenotype data for NanoString nCounter data with at least one column. Each row is one sample, that is the number of rows must equal number of samples or replicates in the data.

**path** path to the csv file.

**header** a logical value indicating whether the file contains the names of the variables as its first line. The default value is TRUE.

**Value**

A NanoStringSet object.

**Methods**

**positiveControl, positiveControl<-** : Access and set positive control genes.

**negativeControl, negativeControl<-** : Access and set negative control genes.

**housekeepingControl, housekeepingControl<-** : Access and set housekeeping control genes.

**positiveFactor, positiveFactor<-** : Access and set positive factors.

**negativeFactor, negativeFactor<-** : Access and set negative factors.

**housekeepingFactor, housekeepingFactor<-** : Access and set housekeeping factors.

**Author(s)**

hong wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**See Also**

positiveControl, negativeControl, housekeepingControl, positiveFactor, negativeFactor, housekeepingFactor

**Examples**

```

endogenous=matrix(rpois(100,50),25,4)
positive=matrix(rpois(24,c(128,32,8,2,0.5,0.125)*80),6,4)
negative=matrix(rpois(32,10),8,4)
housekeeping=matrix(rpois(12,100),3,4)
designs=data.frame(group=c(0,0,1,1),gender=c("male","female","female","male"),
age=c(20,40,39,37))
NanoStringData=createNanoStringSet(endogenous,positive,negative,
                                   housekeeping,designs)

NanoStringData
pData(NanoStringData)
positiveControl(NanoStringData)
head(exprs(NanoStringData))

```

---

negativeControl	<i>Accessor functions for the 'negativeControl' slot in a NanoStringSet object.</i>
-----------------	---

---

**Description**

Negative control genes are provided by nCounter Analyzer which can be used to estimate background noise for each sample.

**Usage**

```

## S4 method for signature 'NanoStringSet'
negativeControl(object)
## S4 replacement method for signature 'NanoStringSet,matrix'
negativeControl(object) <- value

```

**Arguments**

object	A NanoStringSet object.
value	A matrix with negative control genes.

**Details**

Each code set in the nCounter Analyzer includes several negatives control genes for which no tranCounterript is expected to be present. We use these spike-in negative control genes to estimate background noise for each sample.

**Value**

A matrix contain negative control genes

**Author(s)**

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**See Also**

negativeFactor

**Examples**

```

data(NanoStringData)
## obtain negative control genes
negativeControl(NanoStringData)

## assign a matrix
n=ncol(exprs(NanoStringData))
r=nrow(negativeControl(NanoStringData))
negative=matrix(rpois(r*n,10),ncol=n)
negativeControl(NanoStringData)=negative

```

---

negativeFactor	<i>Accessor functions for the 'negativeFactor' slot in a NanoStringSet object.</i>
----------------	--

---

**Description**

Negative size factors can be used to adjust background noise for each sample.

**Usage**

```

## S4 method for signature 'NanoStringSet'
negativeFactor(object)
## S4 replacement method for signature 'NanoStringSet,numeric'
negativeFactor(object) <- value

```

**Arguments**

object	A NanoStringSet object.
value	A vector of background noise.

**Details**

Accurate estimation of system background is essential for DE detection analysis. Each code set in the nCounter Analyzer includes several negative control genes for which no transcript is expected to be present. We use these spike-in negative control genes to estimate background noise for each sample

**Value**

A vector contain background noise

**Author(s)**

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

**See Also**

negativeControl

**Examples**

```
data(NanoStringData)
## obtain negative factors
negativeFactor(NanoStringData)

## assign a vector
n=ncol(exprs(NanoStringData))
lamda=rpois(n,10)
negativeFactor(NanoStringData)=lamda
```

---

PlotsPositiveHousekeeping

*Plots of positive controls and housekeeping genes*

---

**Description**

This function is used to pre-check the expressions of positive controls and housekeeping genes before data analysis. Linear regression plot of positive controls and variation analysis of housekeeping genes are available. The expressions of positive controls are supposed to be linearly related to the concentration of input sample materials, and the expressions of housekeeping genes are supposed to have relatively low variation. Nanostring recommends at least three housekeeping genes, but the more that are included, the more accurate the normalization will be.

**Usage**

```
PlotsPositiveHousekeeping(path=path, header=TRUE)
```

**Arguments**

path	the path of the file which the data are to be read from.
header	a logical value indicating whether the file contains the names of the variables as its first line. If missing, the value is determined from the file format: header is set to TRUE if and only if the first row contains one fewer field than the number of columns.

**Author(s)**

hong wang <hong.wang@uky.edu> tingting zhai <tingting.zhai@uky.edu> chi wang <chi.wang@uky.edu>

**Examples**

```
##path="/Users/NanoStringdiff-Rcode/Data/horbinski.csv"
##PlotsPositiveHousekeeping(path=path, header=TRUE)
```

---

positiveControl	<i>Accessor functions for the 'positiveControl' slot in a NanoStringSet object.</i>
-----------------	---

---

### Description

nCounter Analyzer has positive spike-in RNA hybridization controls for each sample which can be used to estimate the overall efficiency of hybridization and recovery for each sample.

### Usage

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

### Arguments

object	A NanoStringSet object.
value	A matrix with six positive control genes.

### Details

Positive control genes are provided by NanoString nCounter technology. For each sample, nCounter provide six positive controls corresponding to six different concentrations in the 30 ul hybridization: 128fM, 32fM, 8fM, 2fM, 0.5fM, and 0.125fM. Six positive control genes must be order by concentrations from high to low.

### Value

A matrix contain positive control genes

### Author(s)

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

### See Also

positiveFactor

### Examples

```
data(NanoStringData)  
## obtain positive control genes  
positiveControl(NanoStringData)  
  
## assign a matrix  
n=ncol(exprs(NanoStringData))
```

```
x=matrix(c(128,32,8,2,0.5,0.125)*80,ncol=1)
positive=matrix(rpois(6*n,x),ncol=n)
positiveControl(NanoStringData)=positive
```

---

positiveFactor	<i>Accessor functions for the 'positiveFactor' slot in a NanoStringSet object.</i>
----------------	--

---

### Description

Positive size factors can be used to adjust all platform associated sources of variation.

### Usage

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

### Arguments

object	A NanoStringSet object.
value	A vector of positive size factors.

### Details

The observed counts including negative control genes and housekeeping control genes might be affected by some experimental factors like hybridization and binding efficiency. In order to get the true rate of gene expression, these variations must be normalized. Positive size factors can normalize this kind of variation.

### Value

A vector contain positive size factors

### Author(s)

Hong Wang <hong.wang@uky.edu> chi wang <chi.wang@uky.edu>

### See Also

positiveControl

**Examples**

```
data(NanoStringData)
## obtain positive factors
positiveFactor(NanoStringData)

## assign a vector
n=ncol(exprs(NanoStringData))
positiveFactor(NanoStringData)=rep(1,n)
```

# Index

## \* **Plots, Positive Controls, Housekeeping Genes**

PlotsPositiveHousekeeping, 12

## \* **classes**

NanoStringSet-class, 8

## \* **datasets**

NanoStringData, 7

## \* **models**

glm.LRT, 4

## \* **normalization**

estNormalizationFactors, 3

NanoStringDataNormalization, 8

## \* **package**

NanoStringDiff-package, 2

createNanoStringSet

(NanoStringSet-class), 8

createNanoStringSetFromCsv

(NanoStringSet-class), 8

estNormalizationFactors, 3

estNormalizationFactors,estNormalizationFactors-method  
(estNormalizationFactors), 3

glm.LRT, 4

glm.LRT,NanoStringSet-method (glm.LRT),  
4

housekeepingControl, 5

housekeepingControl,NanoStringSet-method  
(housekeepingControl), 5

housekeepingControl<-  
(housekeepingControl), 5

housekeepingControl<-,NanoStringSet,matrix-method  
(housekeepingControl), 5

housekeepingFactor, 6

housekeepingFactor,NanoStringSet-method  
(housekeepingFactor), 6

housekeepingFactor<-

(housekeepingFactor), 6

housekeepingFactor<-,NanoStringSet,numeric-method  
(housekeepingFactor), 6

NanoStringData, 7

NanoStringDataNormalization, 8

NanoStringDataNormalization,NanoStringDataNormalization-me  
(NanoStringDataNormalization),  
8

NanoStringDiff

(NanoStringDiff-package), 2

NanoStringDiff-package, 2

NanoStringSet (NanoStringSet-class), 8

NanoStringSet-class, 8

negativeControl, 10

negativeControl,NanoStringSet-method  
(negativeControl), 10

negativeControl<- (negativeControl), 10

negativeControl<-,NanoStringSet,matrix-method  
(negativeControl), 10

negativeFactor, 11

negativeFactor,NanoStringSet-method  
(negativeFactor), 11

negativeFactor<- (negativeFactor), 11

negativeFactor<-,NanoStringSet,numeric-method  
(negativeFactor), 11

PlotsPositiveHousekeeping, 12

PlotsPositiveHousekeeping,PlotsPositiveHousekeeping-method  
(PlotsPositiveHousekeeping), 12

positiveControl, 13

positiveControl,NanoStringSet-method  
(positiveControl), 13

positiveControl<- (positiveControl), 13

positiveControl<-,NanoStringSet,matrix-method  
(positiveControl), 13

positiveFactor, 14

positiveFactor,NanoStringSet-method  
(positiveFactor), 14

positiveFactor<- (positiveFactor), 14



positiveFactor<- ,NanoStringSet,numeric-method  
(positiveFactor), [14](#)