

# Package: standR (via r-universe)

May 30, 2026

**Title** Spatial transcriptome analyses of Nanostring's DSP data in R

**Version** 1.16.0

**Description** standR is an user-friendly R package providing functions to assist conducting good-practice analysis of Nanostring's GeoMX DSP data. All functions in the package are built based on the SpatialExperiment object, allowing integration into various spatial transcriptomics-related packages from Bioconductor. standR allows data inspection, quality control, normalization, batch correction and evaluation with informative visualizations.

**biocViews** Spatial, Transcriptomics, GeneExpression, DifferentialExpression, QualityControl, Normalization, ExperimentHubSoftware

**License** MIT + file LICENSE

**URL** <https://github.com/DavisLaboratory/standR>

**BugReports** <https://github.com/DavisLaboratory/standR/issues>

**Encoding** UTF-8

**LazyData** false

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Imports** dplyr, SpatialExperiment (>= 1.5.2), SummarizedExperiment, SingleCellExperiment, edgeR, rlang, readr, tibble, ggplot2, tidyr, ruv, limma, patchwork, S4Vectors, Biobase, BiocGenerics, grDevices, stats, methods, ggalluvial, mclustcomp, RUVSeq

**Suggests** knitr, ExperimentHub, rmarkdown, scater, uwot, ggpubr, ggrepel, cluster, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Depends** R (>= 4.1)

**VignetteBuilder** knitr

**Config/pak/sysreqs** make libmagick++-dev gsfontr libbz2-dev libicu-dev libjpeg-dev liblzma-dev libpng-dev libxml2-dev libssl-dev libx11-dev xz-utils zlib1g-dev

**Repository** <https://bioc-release.r-universe.dev>

**Date/Publication** 2026-04-28 12:58:07 UTC

**RemoteUrl** <https://github.com/bioc/standR>

**RemoteRef** RELEASE\_3\_23

**RemoteSha** 0359c71796707dc2e60f789e3b11827b3063d94e

## Contents

addPerROIQC . . . . .	2
computeClusterEvalStats . . . . .	3
drawPCA . . . . .	4
findBestK . . . . .	6
findNCGs . . . . .	7
geomxBatchCorrection . . . . .	8
geomxNorm . . . . .	9
plotClusterEvalStats . . . . .	10
plotDR . . . . .	11
plotGeneQC . . . . .	12
plotMDS . . . . .	14
plotPairPCA . . . . .	16
plotPCAbiplot . . . . .	17
plotRLEExpr . . . . .	18
plotROIQC . . . . .	19
plotSampleInfo . . . . .	20
plotScreePCA . . . . .	21
prepareSpatialDecon . . . . .	22
readGeoMx . . . . .	23
readGeoMxFromDGE . . . . .	24
spe2dge . . . . .	25
<b>Index</b>	<b>26</b>

---

addPerROIQC

*Add QC statistics to the Spatial Experiment object*

---

## Description

Add QC statistics to the Spatial Experiment object

**Usage**

```
addPerROIQC(  
  spe_object,  
  sample_fraction = 0.9,  
  rm_genes = TRUE,  
  min_count = 5,  
  design = NULL  
)
```

**Arguments**

spe_object	A SpatialExperiment object
sample_fraction	Double. Genes with low count in more than this threshold of the samples will be removed. Default is 0.9
rm_genes	Logical. Decide whether genes with low count in more than sample_fraction of the samples are removed from the dataset. Default is TRUE.
min_count	Integer. Minimum read count to calculate count threshold. Default is 5.
design	Generate using model.matrix, if this is specify, edgeR::filterByExpr will be used to filter genes.

**Value**

A SpatialExperiment object

**Examples**

```
data("dkd_spe_subset")  
spe_filtered <- addPerROIQC(dkd_spe_subset)  
spe_filtered
```

---

computeClusterEvalStats

*Calculate statistics for evaluating batch correction*

---

**Description**

Calculate statistics for evaluating batch correction

**Usage**

```
computeClusterEvalStats(
  spe_object,
  foiColumn,
  precomputed = NULL,
  n_dimension = c(1, 2),
  assay = 2
)
```

**Arguments**

spe_object	A Spatial Experiment object.
foiColumn	A column name indicating the factor of interest to be tested, can be biological factor or batch factor.
precomputed	a dimensional reduction results from stats::prcomp. result in reducedDims(object) to plot. Default is NULL, we will compute for you.
n_dimension	The top n dimensions to be plotted
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).

**Value**

A dataframe object containing the clustering evaluating statistics.

**Examples**

```
library(scater)
data("dkd_spe_subset")
computeClusterEvalStats(dkd_spe_subset, "SlideName")
```

---

drawPCA	<i>Compute and plot the results of a PCA analysis on gene expression data</i>
---------	---

---

**Description**

Compute and plot the results of a PCA analysis on gene expression data

**Usage**

```
drawPCA(object, dims = c(1, 2), ...)

## S4 method for signature 'ExpressionSet'
drawPCA(object, dims = c(1, 2), precomputed = NULL, textScale = 1, ...)
```

```

## S4 method for signature 'SummarizedExperiment'
drawPCA(
  object,
  dims = c(1, 2),
  assay = 1,
  precomputed = NULL,
  textScale = 1,
  ...
)

## S4 method for signature 'SingleCellExperiment'
drawPCA(
  object,
  dims = c(1, 2),
  assay = 1,
  precomputed = NULL,
  textScale = 1,
  ...
)

## S4 method for signature 'SpatialExperiment'
drawPCA(
  object,
  dims = c(1, 2),
  assay = 1,
  precomputed = NULL,
  textScale = 1,
  ...
)

```

### Arguments

object	a DGEList, SummarizedExperiment or ExpressionSet object containing gene expression data.
dims	a numeric, containing 2 values specifying the dimensions to plot.
...	aesthetic mappings to pass to <code>ggplot2::aes_string()</code> .
precomputed	a dimensional reduction results from <code>stats::prcomp</code> . result in <code>reducedDims(object)</code> to plot.
textScale	a numeric, specifying the relative scale factor to apply to text on the plot.
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).

### Value

a ggplot2 object

**Examples**

```
data("dkd_spe_subset")
drawPCA(dkd_spe_subset)
```

---

findBestK

*Testing multiple K for RUV4 batch correction to find the best K.*


---

**Description**

Testing multiple K for RUV4 batch correction to find the best K.

**Usage**

```
findBestK(
  spe,
  maxK = 10,
  factor_of_int,
  factor_batch,
  NCGs,
  point_size = 3,
  line_col = "black",
  point_col = "black",
  text_size = 13
)
```

**Arguments**

spe	A Spatial Experiment object.
maxK	Integer. The max k to test, will test k from 1 to maxK, by default is 10.
factor_of_int	Column name(s) to indicate the factors of interest. This is required for the RUV4 method.
factor_batch	Column name to indicate the batch.
NCGs	Negative control genes. This is required for the RUV4 method.
point_size	Numeric. Plotting parameter.
line_col	Character. Plotting parameter.
point_col	Character. Plotting parameter.
text_size	Numeric. Plotting parameter.

**Value**

A ggplot object.

**Examples**

```
data("dkd_spe_subset")
spe <- findNCGs(dkd_spe_subset, top_n = 100)
findBestK(spe,
  factor_of_int = c("disease_status"),
  factor_batch = "SlideName", NCGs = S4Vectors::metadata(spe)$NCGs
)
```

---

findNCGs	<i>Get negative control genes from each batch of the data</i>
----------	---

---

**Description**

Get negative control genes from each batch of the data

**Usage**

```
findNCGs(spe, n_assay = 2, batch_name = "SlideName", top_n = 200)
```

**Arguments**

spe	A Spatial Experiment object.
n_assay	Integer to indicate the nth count table in the assay(spe) to be used.
batch_name	Column name indicating batches.
top_n	Integer indicate how many genes to be included as negative control genes.

**Value**

A Spatial Experiment object, containing negative control genes in the metadata.

**Examples**

```
data("dkd_spe_subset")

spe <- findNCGs(dkd_spe_subset, top_n = 100)
S4Vectors::metadata(spe)$NCGs
```

---

geomxBatchCorrection *Batch correction for GeoMX data*

---

### Description

Batch correction for GeoMX data

### Usage

```
geomxBatchCorrection(
  spe,
  k,
  factors,
  NCGs,
  n_assay = 2,
  batch = NULL,
  batch2 = NULL,
  covariates = NULL,
  design = matrix(1, ncol(spe), 1),
  method = c("RUV4", "Limma", "RUVg"),
  isLog = TRUE,
  outAssay = "logcounts"
)
```

### Arguments

spe	A Spatial Experiment object.
k	The number of unwanted factors to use. Can be 0. This is required for the RUV4 method.
factors	Column name(s) to indicate the factors of interest. This is required for the RUV4 method.
NCGs	Negative control genes. This is required for the RUV4 method.
n_assay	Integer to indicate the nth count table in the assay(spe) to be used.
batch	A vector indicating batches. This is required for the Limma method.
batch2	A vector indicating the second series of batches. This is specific for the Limma method.
covariates	A matrix or vector of numeric covariates to be adjusted for.
design	A design matrix relating to treatment conditions to be preserved, can be generated using <code>stats::model.matrix</code> function with all biological factors included.
method	Can be either RUV4 or Limma or RUVg, by default is RUV4.
isLog	Logical vector, indicating if the count table is log or not.
outAssay	Output assay name, logcounts by default.

**Value**

A Spatial Experiment object, containing the normalized count and normalization factor. For method RUV4 and RUVg, the W matrices will be saved in the colData of the object.

**Note**

The normalised count is not intended to be used directly for linear modelling. For linear modelling, it is better to include the batch factors/W matrices in the linear model.

**References**

Gagnon-Bartsch, J. A., Jacob, L., & Speed, T. P. (2013). Removing unwanted variation from high dimensional data with negative controls. Berkeley: Tech Reports from Dep Stat Univ California, 1-112.

Ritchie, M. E., Phipson, B., Wu, D. I., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic acids research*, 43(7), e47-e47.

**Examples**

```
data("dkd_spe_subset")
spe <- findNCGs(dkd_spe_subset, top_n = 100)
spe_ruv <- geomxBatchCorrection(spe,
  k = 3,
  factors = c("disease_status", "region"),
  NCGs = S4Vectors::metadata(spe)$NCGs
)
```

---

geomxNorm

*Perform normalization to GeoMX data*

---

**Description**

Perform normalization to GeoMX data

**Usage**

```
geomxNorm(
  spe_object,
  method = c("TMM", "RPKM", "TPM", "CPM", "upperquartile", "sizefactor"),
  log = TRUE
)
```

**Arguments**

spe_object	A SpatialExperiment object.
method	Normalization method to use. Options: TMM, RPKM, TPM, CPM, upperquartile, sizefactor. RPKM and TPM require gene length information, which should be added into rowData(spe). Note that TMM here is TMM + CPM.
log	Log-transformed or not.

**Value**

A SpatialExperiment object, with the second assay being the normalized count matrix. The normalised count is stored in the assay slot called "logcounts" by default. With method TMM and sizefactor, the norm.factor will be saved in the metadata of the SpatialExperiment object.

**Note**

The normalised count is not intended to be used directly for linear modelling. For linear modelling, it is better to include the normalized factors in the "norm.factors" column of the DGEList object.

**References**

- Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1), 139-140.
- Love, M., Anders, S., & Huber, W. (2014). Differential analysis of count data—the DESeq2 package. *Genome Biol*, 15(550), 10-1186.

**Examples**

```
data("dkd_spe_subset")

spe_tmm <- geomxNorm(dkd_spe_subset, method = "TMM")
spe_upq <- geomxNorm(dkd_spe_subset, method = "upperquartile")
spe_deseqnorm <- geomxNorm(dkd_spe_subset, method = "sizefactor")
```

---

plotClusterEvalStats *Compare and evaluate different batch corrected data with plotting.*

---

**Description**

Compare and evaluate different batch corrected data with plotting.

**Usage**

```
plotClusterEvalStats(  
  spe_list,  
  bio_feature_name,  
  batch_feature_name,  
  data_names,  
  colors = NA  
)
```

**Arguments**

spe_list	A list of Spatial Experiment object.
bio_feature_name	The common biological variation name.
batch_feature_name	The common batch variation name.
data_names	Data names.
colors	Color values of filling the bars.

**Value**

A ggplot object.

**Examples**

```
library(scater)  
data("dkd_spe_subset")  
spe <- dkd_spe_subset  
spe2 <- spe  
spe3 <- spe  
plotClusterEvalStats(list(spe, spe2, spe3),  
  bio_feature_name = "region",  
  batch_feature_name = "SlideName", c("test1", "test2", "test3")  
)
```

---

plotDR	<i>Compute and plot the results of any dimension reduction methods on gene expression data</i>
--------	--

---

**Description**

Compute and plot the results of any dimension reduction methods on gene expression data

**Usage**

```
plotDR(object, dims = c(1, 2), ...)

## S4 method for signature 'SingleCellExperiment'
plotDR(object, dims, dimred = "PCA", textScale = 1, ...)

## S4 method for signature 'SpatialExperiment'
plotDR(object, dims, dimred = "PCA", textScale = 1, ...)
```

**Arguments**

object	a DGEList, SummarizedExperiment or ExpressionSet object containing gene expression data.
dims	a numeric, containing 2 values specifying the dimensions to plot.
...	aesthetic mappings to pass to <code>ggplot2::aes_string()</code> .
dimred	a string or integer scalar indicating the reduced dimension result in <code>reducedDims(object)</code> to plot.
textScale	a numeric, specifying the relative scale factor to apply to text on the plot.

**Value**

a ggplot2 object

**Examples**

```
library(scater)
data("dkd_spe_subset")
spe <- scater::runPCA(dkd_spe_subset)
plotDR(spe, dimred = "PCA")
```

---

plotGeneQC

*Plot gene-wise QC plot*

---

**Description**

Plot gene-wise QC plot

**Usage**

```
plotGeneQC(
  spe,
  top_n = 9,
  ordannots = c(),
  point_size = 1,
  line_type = "dashed",
```

```

    line_col = "darkred",
    line_cex = 1,
    hist_col = "black",
    hist_fill = "skyblue",
    bin_num = 30,
    text_size = 13,
    layout_ncol = 1,
    layout_nrow = 2,
    layout_height = c(1, 1),
    ...
  )

```

### Arguments

spe	A SpatialExperiment object.
top_n	Integer. Indicating the top n genes will be plotted. Default is 9.
ordannots	variables or computations to sort samples by (tidy style).
point_size	Numeric. Point size.
line_type	Character. Line types for ggplot.
line_col	Color for line.
line_cex	Cex for line.
hist_col	Color for histogram.
hist_fill	Fill for histogram.
bin_num	Bin numbers for histogram.
text_size	Text size.
layout_ncol	Integer. Column number for layout. Default is 1.
layout_nrow	Integer. Row number for layout. Default is 2.
layout_height	Vector of numerics with length of 2. Default is c(1, .4).
...	aesthetic mappings to pass to ggplot2::aes() of the dot plots.

### Value

A ggplot object

### Examples

```

data("dkd_spe_subset")
spe <- addPerROIQC(dkd_spe_subset)
plotGeneQC(spe)

```

---

plotMDS	<i>Compute and plot the results of a MDS analysis on gene expression data</i>
---------	---

---

**Description**

Compute and plot the results of a MDS analysis on gene expression data

**Usage**

```
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)  
  
## S4 method for signature 'DGEList'  
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)  
  
## S4 method for signature 'ExpressionSet'  
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)  
  
## S4 method for signature 'SummarizedExperiment'  
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)
```

```
)  
  
## S4 method for signature 'SingleCellExperiment'  
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)  
  
## S4 method for signature 'SpatialExperiment'  
plotMDS(  
  object,  
  dims = c(1, 2),  
  precomputed = NULL,  
  textScale = 1,  
  assay = 1,  
  ...  
)
```

### Arguments

object	a DGEList, SummarizedExperiment or ExpressionSet object containing gene expression data.
dims	a numeric, containing 2 values specifying the dimensions to plot.
precomputed	a dimensional reduction results from either <code>limma::plotMDS</code> .
textScale	a numeric, specifying the relative scale factor to apply to text on the plot.
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).
...	aesthetic mappings to pass to <code>ggplot2::aes_string()</code> .

### Value

a ggplot2 object

### Examples

```
data("dkd_spe_subset")  
standR::plotMDS(dkd_spe_subset)
```

---

plotPairPCA *Plot pair-wise PCA plots for multiple dimensions*

---

### Description

Plot pair-wise PCA plots for multiple dimensions

### Usage

```
plotPairPCA(
  spe_object,
  n_dimension = 3,
  precomputed = NULL,
  assay = 2,
  title = NA,
  title.size = 14,
  rmduplabs = FALSE,
  flipcoord = FALSE,
  legend.pos = "top",
  ...
)
```

### Arguments

spe_object	A SpatialExperiment object.
n_dimension	The top n dimensions to be plotted
precomputed	a dimensional reduction results from stats::prcomp. result in reducedDims(object) to plot. Default is NULL, we will compute for you.
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).
title	Character vector, title to put at the top.
title.size	Numeric vector, size of the title.
rmduplabs	Remove duplicated labels from the plot. FALSE by default.
flipcoord	Flip the xy coordinates. FALSE by default.
legend.pos	Position of the legend. top by default.
...	aesthetic mappings to pass to ggplot2::aes().

### Value

A ggplot object.

### Examples

```
data("dkd_spe_subset")
plotPairPCA(dkd_spe_subset)
```

---

plotPCAbiplot	<i>Plot PCA bi plot</i>
---------------	-------------------------

---

**Description**

Plot PCA bi plot

**Usage**

```
plotPCAbiplot(
  spe_object,
  n_loadings = 10,
  dims = c(1, 2),
  precomputed = NULL,
  assay = 1,
  arrow_x = 0,
  arrow_y = 0,
  ...
)
```

**Arguments**

spe_object	A SpatialExperiment object.
n_loadings	Plot the top n gene loadings
dims	The top n dimensions to be plotted
precomputed	a dimensional reduction results from stats::prcomp. result in reducedDims(object) to plot. Default is NULL, we will compute for you.
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).
arrow_x	a numeric, indicating the x coordinate of the base of the arrow.
arrow_y	a numeric, indicating the y coordinate of the base of the arrow.
...	aesthetic mappings to pass to ggplot2::aes().

**Value**

A ggplot object.

**Examples**

```
data("dkd_spe_subset")
plotPCAbiplot(dkd_spe_subset)
```

---

plotRLEExpr	<i>Compute and plot relative log expression (RLE) values of gene expression data</i>
-------------	--

---

## Description

Compute and plot relative log expression (RLE) values of gene expression data

## Usage

```
plotRLEExpr(object, ordannots = c(), ...)  
  
## S4 method for signature 'DGEList'  
plotRLEExpr(object, ordannots = c(), ...)  
  
## S4 method for signature 'ExpressionSet'  
plotRLEExpr(object, ordannots = c(), ...)  
  
## S4 method for signature 'SummarizedExperiment'  
plotRLEExpr(object, ordannots, assay = 1, sce_thresh = 1000, ...)
```

## Arguments

object	a DGEList, SummarizedExperiment or ExpressionSet object containing gene expression data.
ordannots	variables or computations to sort samples by (tidy style).
...	aesthetic mappings to pass to <code>ggplot2::aes_string()</code> .
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).
sce_thresh	Integer value. The threshold of sample size for using dot plot instead of box plot.

## Value

a ggplot2 object, containing the RLE plot.

## Examples

```
data("dkd_spe_subset")  
plotRLEExpr(dkd_spe_subset)
```

---

plotROIQC	<i>Plot Sample-wise QC plot</i>
-----------	---------------------------------

---

**Description**

Plot Sample-wise QC plot

**Usage**

```
plotROIQC(
  spe_object,
  x_axis = "AOINucleiCount",
  y_axis = "lib_size",
  x_lab = "AOINucleiCount",
  y_lab = "Library size",
  x_threshold = NULL,
  y_threshold = NULL,
  regression_col = "purple",
  hist_col = "black",
  hist_fill = "white",
  bin_num = 50,
  threshold_col = "red",
  threshold_linetype = "dashed",
  layout_ncol = 2,
  layout_nrow = 2,
  layout_height = c(0.8, 2.5),
  layout_width = c(2.5, 0.8),
  ...
)
```

**Arguments**

spe_object	A SpatialExperiment object.
x_axis	Numeric feature to plot as x axis.
y_axis	Numeric feature to plot as y axis.
x_lab	Label name for x axis.
y_lab	Label name for y axis.
x_threshold	Threshold to draw.
y_threshold	Threshold to draw.
regression_col	Color for the regression line.
hist_col	Color for the histograms.
hist_fill	Fill for the histograms.
bin_num	Bin numbers for the histograms.
threshold_col	Threshold line color.

threshold\_linetype Threshold line type.  
 layout\_ncol Column number layout.  
 layout\_nrow Row number layout.  
 layout\_height Height layout.  
 layout\_width Width layout.  
 ... aesthetic mappings to pass to `ggplot2::aes()` of the dot plots.

**Value**

A ggplot object.

**Examples**

```
library(ggplot2)
library(patchwork)
data("dkd_spe_subset")
spe <- addPerROIQC(dkd_spe_subset)

plotROIQC(spe)
```

---

plotSampleInfo *Plot the user-defined meta data using alluvium plot*

---

**Description**

Plot the user-defined meta data using alluvium plot

**Usage**

```
plotSampleInfo(spe_object, column2plot, textsize = 3)
```

**Arguments**

spe\_object A SpatialExperiment object.  
 column2plot Which columns to plot.  
 textsize text size.

**Value**

A ggplot object

**Examples**

```
library(ggalluvial)

data("dkd_spe_subset")
plotSampleInfo(dkd_spe_subset, column2plot = c("SlideName", "disease_status", "region"))
```

---

plotScreePCA

*Plot the PCA scree plot.*


---

**Description**

Plot the PCA scree plot.

**Usage**

```
plotScreePCA(
  spe_object,
  dims = ncol(spe_object),
  precomputed = NULL,
  assay = 1,
  bar_color = "black",
  bar_fill = "royalblue",
  bar_width = 0.8,
  point_col = "tomato3",
  line_col = "tomato3",
  point_size = 2
)
```

**Arguments**

spe_object	A SpatialExperiment object.
dims	The top n dimensions to be plotted
precomputed	a dimensional reduction results from stats::prcomp. result in reducedDims(object) to plot. Default is NULL, we will compute for you.
assay	a numeric or character, specifying the assay to use (for SummarizedExperiment and its derivative classes).
bar_color	Color for bar.
bar_fill	Fill for bar.
bar_width	Bar width.
point_col	Color for point.
line_col	Color for line.
point_size	Point size.

**Value**

A ggplot object.

**Examples**

```
data("dkd_spe_subset")
plotScreePCA(dkd_spe_subset, dims = 10)
```

---

prepareSpatialDecon	<i>Preparing the inputs for SpatialDecon for doing deconvolution on spatial data</i>
---------------------	--

---

**Description**

Preparing the inputs for SpatialDecon for doing deconvolution on spatial data

**Usage**

```
prepareSpatialDecon(
  spe,
  assay2use = "logcounts",
  negProbeName = "NegProbe-WTX",
  pool = NA
)
```

**Arguments**

spe	SpatialExperiment object.
assay2use	The name of the assay to use. By default is logcounts.
negProbeName	The name of the negative probe gene. By default is NegProbe-WTX.
pool	A vector indicates the pools of the genes. This is required when there are more than one Negative Probes.

**Value**

A list of two dataframes. The first data.frame is the normalised count, the second data.frame is the background for the data.

**Examples**

```
library(ExperimentHub)
eh <- ExperimentHub()

query(eh, "standR")
countFile <- eh[["EH7364"]]
sampleAnnoFile <- eh[["EH7365"]]
```

```
spe <- readGeoMx(countFile, sampleAnnoFile, rmNegProbe = FALSE)
out <- prepareSpatialDecon(spe)
```

---

readGeoMx	<i>Import GeoMX DSP data into a spatial experiment object from file paths</i>
-----------	---

---

### Description

Import GeoMX DSP data into a spatial experiment object from file paths

### Usage

```
readGeoMx(
  countFile,
  sampleAnnoFile,
  featureAnnoFile = NA,
  rmNegProbe = TRUE,
  NegProbeName = "NegProbe-WTX",
  colnames.as.rownames = c("TargetName", "SegmentDisplayName", "TargetName"),
  coord.colnames = c("ROICoordinateX", "ROICoordinateY")
)
```

### Arguments

countFile	tsv file or a dataframe object. Count matrix, with samples in columns and features/genes in rows. The first column is gene names/ids.
sampleAnnoFile	tsv file or a dataframe object. Sample annotations.
featureAnnoFile	tsv file or a dataframe object. Feature/Gene annotations.
rmNegProbe	Logical. Default is TRUE, indicating there are negative probe genes in the data.
NegProbeName	Character. Name of negative probe genes, default is NegProbe-WTX.
colnames.as.rownames	Vector of characters, length of 3. Column names used to capture gene names, sample names and gene names in countFile, sampleAnnoFile and featureAnnoFile, respectively.
coord.colnames	Vector of characters, length of 2. Column names used to capture ROI coordinates.

### Value

A SpatialExperiment object.

**Examples**

```
library(ExperimentHub)

eh <- ExperimentHub()
query(eh, "standR")
countFile <- eh[["EH7364"]]
sampleAnnoFile <- eh[["EH7365"]]

spe <- readGeoMx(countFile, sampleAnnoFile, rmNegProbe = FALSE)
```

---

readGeoMxFromDGE	<i>Import GeoMX DSP data into a spatial experiment object from DGE-List object</i>
------------------	--

---

**Description**

Import GeoMX DSP data into a spatial experiment object from DGEList object

**Usage**

```
readGeoMxFromDGE(dge_object, spatialCoord = NULL)
```

**Arguments**

dge_object	a DGEList object (created using edgeR::DGEList).
spatialCoord	a matrix with coordinates of samples, rowname must be consistent with the column names of dge_object.

**Value**

A SpatialExperiment object.

**Examples**

```
# making a simple DGEList object
ng <- 1000
ns <- 10
Counts <- matrix(rnbinom(ng * ns, mu = 5, size = 2), ng, ns)
rownames(Counts) <- seq(ng)
y <- edgeR::DGEList(counts = Counts, group = rep(seq(2), each = 5))

# transfer into spatial experiment object
coords <- matrix(rnorm(2 * ns), 10, 2)
spe <- readGeoMxFromDGE(dge_object = y, spatialCoord = coords)
spe
```

---

spe2dge	<i>Transfer SpatialExperiment object into DGEList object for DE analysis</i>
---------	--

---

**Description**

Transfer SpatialExperiment object into DGEList object for DE analysis

**Usage**

```
spe2dge(spe)
```

**Arguments**

spe                    SpatialExperiment object.

**Value**

A DGEList.

**Examples**

```
data("dkd_spe_subset")
```

```
spe_tmm <- geomxNorm(dkd_spe_subset, method = "TMM")  
dge <- spe2dge(spe_tmm)
```

# Index

addPerROIQC, [2](#)  
computeClusterEvalStats, [3](#)  
drawPCA, [4](#)  
drawPCA, DGEList-method (drawPCA), [4](#)  
drawPCA, ExpressionSet-method (drawPCA), [4](#)  
drawPCA, SingleCellExperiment-method (drawPCA), [4](#)  
drawPCA, SpatialExperiment-method (drawPCA), [4](#)  
drawPCA, SummarizedExperiment-method (drawPCA), [4](#)  
  
findBestK, [6](#)  
findNCGs, [7](#)  
  
geomxBatchCorrection, [8](#)  
geomxNorm, [9](#)  
  
plotClusterEvalStats, [10](#)  
plotDR, [11](#)  
plotDR, SingleCellExperiment-method (plotDR), [11](#)  
plotDR, SpatialExperiment-method (plotDR), [11](#)  
plotGeneQC, [12](#)  
plotMDS, [14](#)  
plotMDS, DGEList-method (plotMDS), [14](#)  
plotMDS, ExpressionSet-method (plotMDS), [14](#)  
plotMDS, SingleCellExperiment-method (plotMDS), [14](#)  
plotMDS, SpatialExperiment-method (plotMDS), [14](#)  
plotMDS, SummarizedExperiment-method (plotMDS), [14](#)  
plotPairPCA, [16](#)  
plotPCAbiplot, [17](#)  
plotRLEExpr, [18](#)  
plotRLEExpr, DGEList-method (plotRLEExpr), [18](#)  
plotRLEExpr, ExpressionSet-method (plotRLEExpr), [18](#)  
plotRLEExpr, SummarizedExperiment-method (plotRLEExpr), [18](#)  
plotROIQC, [19](#)  
plotSampleInfo, [20](#)  
plotScreePCA, [21](#)  
prepareSpatialDecon, [22](#)  
  
readGeoMx, [23](#)  
readGeoMxFromDGE, [24](#)  
  
spe2dge, [25](#)