

Package: gscreend (via r-universe)

October 1, 2026

Type Package

Title Analysis of pooled genetic screens

Version 1.27.0

Description Package for the analysis of pooled genetic screens (e.g. CRISPR-KO). The analysis of such screens is based on the comparison of gRNA abundances before and after a cell proliferation phase. The gscreend packages takes gRNA counts as input and allows detection of genes whose knockout decreases or increases cell proliferation.

License GPL-3

Encoding UTF-8

LazyData false

Depends R (>= 3.6)

Imports SummarizedExperiment, nloptr, fGarch, methods, BiocParallel, graphics

Suggests knitr, testthat, rmarkdown, BiocStyle

VignetteBuilder knitr

RoxygenNote 7.2.3

biocViews Software, StatisticalMethod, PooledScreens, CRISPR

URL <https://github.com/imkeller/gscreend>

BugReports <https://github.com/imkeller/gscreend/issues>

Config/pak/sysreqs cmake zlib1g-dev

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

Date/Publication 2026-04-28 12:51:08 UTC

RemoteUrl <https://github.com/bioc/gscreend>

RemoteRef HEAD

RemoteSha b8621dfc5a485a070817ca8419ec6bfe9cf6958f

Contents

createPoolScreenExp	2
GeneData	3
GeneData,PoolScreenExp-method	3
plotModelParameters	4
plotReplicateCorrelation	5
PoolScreenExp-class	5
ResultsTable	6
RunGscreen	7
sgRNAData	8
sgRNAData,PoolScreenExp-method	8
Index	10

createPoolScreenExp	<i>Create PoolScreenExp Experiment</i>
---------------------	--

Description

Create PoolScreenExp Experiment

Usage

createPoolScreenExp(data)

Arguments

data Input data object containing gRNA level data (SummarizedExperiment)

Value

object PoolScreenExp object

Examples

```
raw_counts <- read.table(
  system.file('extdata', 'simulated_counts.txt',
    package = 'gscreen'),
  header=TRUE)

counts_matrix <- cbind(raw_counts$library0, raw_counts$R0_0, raw_counts$R1_0)

rowData <- data.frame(sgRNA_id = raw_counts$sgrna_id,
  gene = raw_counts$Gene)

colData <- data.frame(samplename = c('library', 'R1', 'R2'),
  timepoint = c('T0', 'T1', 'T1'))

library(SummarizedExperiment)
```

```
se <- SummarizedExperiment(assays=list(counts=counts_matrix),
rowData=rowData, colData=colData)

# create a PoolScreenExp experiment
pse <- createPoolScreenExp(se)
```

GeneData

*GeneData: set and retrieve GeneData of PoolScreenExp***Description**

GeneData: set and retrieve GeneData of PoolScreenExp

Usage

```
GeneData(x)
```

Arguments

x PoolScreenExp object

Value

Gene slot of the object

Examples

```
# import a PoolScreenExp object that has been generated using
# RunGscreen()
pse_an <- readRDS(
system.file('extdata', 'gscreend_analysed_experiment.RData',
package = 'gscreend'))

GeneData(pse_an)
```

GeneData, PoolScreenExp-method

*Accessor function for the Gene slot of the PoolScreenExp class***Description**

Accessor function for the Gene slot of the PoolScreenExp class

Usage

```
## S4 method for signature 'PoolScreenExp'  
GeneData(x)
```

Arguments

x PoolScreenExp object

Value

Gene slot of the object

Examples

```
# import a PoolScreenExp object that has been generated using  
# RunGscreenend()  
pse_an <- readRDS(  
  system.file('extdata', 'gscreenend_analysed_experiment.RData',  
  package = 'gscreenend'))  
  
GeneData(pse_an)
```

plotModelParameters *Plot model parameters from the fitting*

Description

Plot model parameters from the fitting

Usage

```
plotModelParameters(object)
```

Arguments

object PoolScreenExp object

Value

plot

Examples

```
# import a PoolScreenExp object that has been generated using  
# RunGscreenend()  
pse_an <- readRDS(  
  system.file('extdata', 'gscreenend_analysed_experiment.RData',  
  package = 'gscreenend'))  
plotModelParameters(pse_an)
```

```
plotReplicateCorrelation
```

Plot replicate correlation

Description

Plot replicate correlation

Usage

```
plotReplicateCorrelation(object, rep1 = "R1", rep2 = "R2")
```

Arguments

object	PoolScreenExp object
rep1	Name of replicate 1
rep2	Name of replicate 2

Value

replicate_plot

Examples

```
# import a PoolScreenExp object that has been generated using RunGscreen()
pse_an <- readRDS(
  system.file('extdata', 'gscreen_analysed_experiment.RData',
    package = 'gscreen'))
plotReplicateCorrelation(pse_an, rep1 = 'R1', rep2 = 'R2')
```

```
PoolScreenExp-class
```

Class to store pooled CRISPR screening experiment

Description

The poolScreenExp class is an S4 class used to store sgRNA and gene related data as well as parameters necessary for statistical model.

Slots

`sgRNAData` A SummarizedExperiment containing the data related to sgRNAs.

`FittingIntervals` A vector defining the limits of the intervals used for fitting of null model.

`LFCModelParameters` A vector of parameters estimated when fitting the null model.

`GeneData` SummarizedExperiment containing the data related to genes.

`FittingOptions` A named list with options for fitting: `IntervalFraction` - fraction of sgRNAs used in every fitting interval (default 0.1), `alphaCutoff` - alpha cutoff for alpha RRA algorithm (default: 0.05).

ResultsTable	<i>Extract a results table</i>
--------------	--------------------------------

Description

Extract a results table

Usage

```
ResultsTable(object, direction = "negative")
```

Arguments

<code>object</code>	PoolScreenExp object
<code>direction</code>	Whether the table should contain information on positive or negative fold changes ['negative' 'positive']

Value

plot

Examples

```
# import a PoolScreenExp object that has been generated using
# RunGscreen()
pse_an <- readRDS(
  system.file('extdata', 'gscreen_analysed_experiment.RData',
    package = 'gscreen'))
ResultsTable(pse_an, direction = 'negative')
```

RunGscreen	<i>run gscreen</i>
------------	--------------------

Description

run gscreen

Usage

```
RunGscreen(object, quant1 = 0.1, quant2 = 0.9, alphacutoff = 0.05)
```

Arguments

object	PoolScreenExp object
quant1	lower quantile for least quantile of squares regression (default: 0.1)
quant2	upper quantile for least quantile of squares regression (default: 0.9)
alphacutoff	alpha cutoff for alpha-RRA (default: 0.05)

Value

object

Examples

```
raw_counts <- read.table(  
  system.file('extdata', 'simulated_counts.txt',  
  package = 'gscreen'),  
  header=TRUE)  
  
# Create the PoolScreenExp to be analyzed  
counts_matrix <- cbind(raw_counts$library0, raw_counts$R0_0, raw_counts$R1_0)  
  
rowData <- data.frame(sgRNA_id = raw_counts$sgRNA_id,  
  gene = raw_counts$Gene)  
  
colData <- data.frame(samplename = c('library', 'R1', 'R2'),  
  timepoint = c('T0', 'T1', 'T1'))  
  
library(SummarizedExperiment)  
se <- SummarizedExperiment(assays=list(counts=counts_matrix),  
  rowData=rowData, colData=colData)  
  
pse <- createPoolScreenExp(se)  
  
# Run Analysis  
pse_an <- RunGscreen(pse)
```

sgRNAData

sgRNAData: set and retrieve sgRNAData of PoolScreenExp

Description

sgRNAData: set and retrieve sgRNAData of PoolScreenExp

Usage

```
sgRNAData(x)
```

Arguments

x PoolScreenExp object

Value

sgRNA slot of the object

Examples

```
# import a PoolScreenExp object that has been generated using
# RunGscreen()
pse_an <- readRDS(
  system.file('extdata', 'gscreend_analysed_experiment.RData',
    package = 'gscreend'))

sgRNAData(pse_an)
```

sgRNAData, PoolScreenExp-method

Accessor function for the sgRNA slot of the PoolScreenExp class

Description

Accessor function for the sgRNA slot of the PoolScreenExp class

Usage

```
## S4 method for signature 'PoolScreenExp'
sgRNAData(x)
```

Arguments

x PoolScreenExp object

Value

sgRNA slot of the object

Examples

```
# import a PoolScreenExp object that has been generated using
# RunGscreen()
pse_an <- readRDS(
  system.file('extdata', 'gscreen_analysed_experiment.RData',
    package = 'gscreen'))

sgRNAData(pse_an)
```

Index

`createPoolScreenExp`, [2](#)

`GeneData`, [3](#)

`GeneData`, `PoolScreenExp`-method, [3](#)

`plotModelParameters`, [4](#)

`plotReplicateCorrelation`, [5](#)

`PoolScreenExp`-class, [5](#)

`ResultsTable`, [6](#)

`RunGscreen`, [7](#)

`sgRNAData`, [8](#)

`sgRNAData`, `PoolScreenExp`-method, [8](#)