Package ‘synlet’

May 30, 2024

Type Package

Title Hits Selection for Synthetic Lethal RNAi Screen Data

Version 2.4.0

Description Select hits from synthetic lethal RNAi screen data. For example, there are two identical celllines except one gene is knocked-down in one cellline. The interest is to find genes that lead to stronger lethal effect when they are knocked-down further by siRNA. Quality control and various visualisation tools are implemented. Four different algorithms could be used to pick up the interesting hits. This package is designed based on 384 wells plates, but may apply to other platforms with proper configuration.

License GPL-3

biocViews ImmunoOncology, CellBasedAssays, QualityControl, Preprocessing, Visualization, FeatureExtraction

Imports data.table, ggplot2, grDevices, magrittr, methods, patchwork, RankProd, RColorBrewer, stats, utils

Suggests BiocStyle, knitr, testthat, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 7.2.3

Encoding UTF-8

git_url https://git.bioconductor.org/packages/synlet

git_branch RELEASE_3_19

git_last_commit ff31b8a

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29

Author Chunxuan Shao [aut, cre]

Maintainer Chunxuan Shao <chunxuan@outlook.com>
**bScore**

## Contents

- `bScore` ......................................................... 2
- `example_dt` ............................................. 3
- `madSelect` .............................................. 4
- `plateHeatmap` ......................................... 5
- `rankProdHits` ........................................... 6
- `rsaHits` .................................................. 7
- `scatterPlot` ............................................. 8
- `siRNAPlot` .............................................. 9
- `tTest` ..................................................... 10
- `zFactor` .................................................. 11

Index .............................. 12

---

### bScore

*Calculate B-score*

#### Description

Calculate the B-score for plates belonging to the same master plate. Positive / negative controls are removed from the calculation.

#### Usage

```
bScore(masterPlate, dta, treatment, control, outFile = FALSE)
```

#### Arguments

- `masterPlate` a master plate to be normalized.
- `dta` synthetic lethal RNAi screen data.
- `treatment` the treatment experiment condition in EXPERIMENT_MODIFICATION
- `control` the control experiment condition in EXPERIMENT_MODIFICATION.
- `outFile` should calculated B-score files be written to the current folder? File names is (masterPlate).bscore.csv.

#### Value

A list contains B-score for each master plate, treatment plates are the first columns, followed by control plates.

#### References

Examples

```r
data(example_dt)
res <- sapply(unique(example_dt$MASTER_PLATE), bScore, example_dt,
              treatment = "treatment", control = "control", simplify = FALSE)
```

Description

A dataset containing synthetic lethal RNAi screen data to show how functions work. The variables are as follows (all are character except READOUT):

Usage

```r
data(example_dt)
```

Format

A data.table with 4320 rows and 8 variables

Details

- PLATE. plate names.
- MASTER_PLATE. master plate names.
- WELL_CONTENT_NAME. siRNA targets of wells.
- EXPERIMENT_TYPE. sample, negative/positive controls.
- EXPERIMENT_MODIFICATION. experiment conditions, "treatment" or "control".
- ROW_NAME. row names of plates.
- COL_NAME. column names of plates.
- READOUT. screen results.

Value

A data.table containing RNAi screen data, the READOUT value has no real biological meaning.
madSelect  
*Select hits basing on median +/- k*MAD*

**Description**

Select hits basing on median +/- k*MAD, by default k is three.

**Usage**

```r
madSelect(
  masterPlate,
  dat,
  k = 3,
  treatment,
  control,
  outFile = FALSE,
  normMethod = "PLATE"
)
```

**Arguments**

- `masterPlate`  
  the master plate to analysis
- `dat`  
  synthetic lethal RNAi screen data
- `k`  
  cutoff for selecting hits, default is three
- `treatment`  
  the treatment condition in EXPERIMENT_MODIFICATION
- `control`  
  the control condition in EXPERIMENT_MODIFICATION
- `outFile`  
  whether or not write the median normalized results
- `normMethod`  
  normalization methods to be used. If "PLATE", the raw readouts are normalized by plate median, otherwise use median provided control siRNA.

**Value**

A data.frame contains the hits selection results.

- `MASTER_PLATE`: location of siRNA
- `treat_cont_ratio`: ratio of treatment / control
- `treat_median`: median value of treatment plates
- `control_median`: median value of control plates
- `Hits`: Is this siRNA a hit?

**References**

plateHeatmap

Examples

```r
data(example_dt)
res <- sapply((unique(example_dt$MASTER_PLATE)),
               madSelect,
               example_dt,
               control = "control",
               treatment = "treatment",
               simplify = FALSE)
```

plateHeatmap  

Heatmap of all plates

Description

Put all individual plates in one graph, values are the readout in experiments.

Usage

```r
plateHeatmap(dta, base_size = 12, heatmap_col = NULL)
```

Arguments

- `dta` synthetic lethal RNAi screen data
- `base_size` basic font size used for x/y axis and title for heatmaps
- `heatmap_col` color function generated by colorRampPalette.

Value

- a ggplot object

Examples

```r
data(example_dt)
plateHeatmap(example_dt)
```
Select hits by the rank product method

Description
Select hits by rank product methods by comparing treatment and control.

Usage
`rankProdHits(masterPlate, dta, treatment, control, normMethod = "PLATE")`

Arguments
- `masterPlate`: the master plate to be analyzed
- `dta`: synthetic lethal RNAi screen data
- `treatment`: the treatment condition in EXPERIMENT_MODIFICATION
- `control`: the control condition in EXPERIMENT_MODIFICATION
- `normMethod`: normalization methods to be used. If "PLATE", the raw readouts are normalized by plate median, otherwise use provided control siRNA

Value
A list contains results by the rank product method for each master plate.

- **MASTER_PLATE**: location of siRNA
- **pvalue_treat_lowerthan_cont**: p-value for the hypothesis that treatment has lower normalized readout compared to control
- **FDR_treat_lowerthan_cont**: FDR value
- **treat_cont_log2FC**: log2 fold change of treatment / control

References

Examples
```r
data(example_dt)
res <- sapply(unique(example_dt$MASTER_PLATE),
              rankProdHits,
              example_dt,
              control = "control",
              treatment = "treatment",
              simplify = FALSE)
```
Description

Selected hits by redundant siRNA activity method. Here is a wrapper function of RSA 1.8 by Yingyao Zhou.

Usage

rsaHits(
  dta,
  treatment,
  control,
  normMethod = "PLATE",
  LB,
  UB,
  revHits = FALSE,
  Bonferroni = FALSE,
  outputFile = "RSAhits.csv",
  scoreFile = "RSA_score.csv"
)

Arguments

da  synthetic lethal RNAi screen data

treatment  the treatment condition in EXPERIMENT_MODIFICATION

control  the control condition in EXPERIMENT_MODIFICATION

normMethod  normalization methods. If "PLATE", then values are normalized by plate median, otherwise use the provided control siRNA

LB  Low bound

UB  up bound

revHits  reverse hit picking, default the lower the score the better

Bonferroni  conceptually useful when there are different number of siRNAs per gene, default FALSE

outputFile  output file name

scoreFile  name of the score file to be written under the current folder

Value

A result file written to the current folder.

- Gene_ID,Well_ID,Score: columns from input spreadsheet
- LogP: OPI p-value in log10, i.e., -2 means 0.01
• **OPI_Hit**: whether the well is a hit, 1 means yes, 0 means no
• **#hitWell**: number of hit wells for the gene
• **#totalWell**: total number of wells for the gene. If gene A has three wells w1, w2 and w3, and w1 and w2 are hits, #totalWell should be 3, #hitWell should be 2, w1 and w2 should have OPI_Hit set as 1 and w3 should have OPI_Hit set as 0.
• **OPI_Rank**: ranking column to sort all wells for hit picking
• **Cutoff_Rank**: ranking column to sort all wells based on Score in the simple activity-based method

Note: a rank value of 999999 means the well is not a hit

**References**


**Examples**

```r
data(example_dt)
rsaHits(example_dt, treatment = "treatment", control = "control",
normMethod = "PLATE", LB = 0.2, UB = 0.8, revHits = FALSE,
Bonferroni = FALSE, outputFile = "RSAhits.csv")
```

---

**scatterPlot**  
*Scatter plot of RNAi screen results*

**Description**

Produce a single plot for readouts of each plate, with the option of highlighting specific signals, like positive/negative controls.

**Usage**

```r
scatterPlot(
  dta,
  scatter_colours = rainbow(10),
  controlOnly = FALSE,
  control_name = NULL
)
```

**Arguments**

- `dta`  
  synthetic lethal RNAi screen data
- `scatter_colours`  
  colour for different signals
- `controlOnly`  
  whether or not to plot control wells only
- `control_name`  
  names of control siRNAs.
siRNAPlot

Value

a ggplot object

Examples

data(example_dt)
scatterPlot(example_dt, control_name = c("PLK1 si1", "scrambled control si1", "lipid only"))

siRNAPlot  Plot siRNA data and quality metrics.

Description

Plot the normalized RNAi screen data, row data, control signals and Z’ factor.

Usage

siRNAPlot(
  gene,
  dta,
  controlsiRNA,
  FILEPATH = ".",
  colour = rainbow(10),
  zPrimeMed,
  zPrimeMean,
  treatment,
  control,
  normMethod = c("PLATE"),
  save_plot = FALSE,
  width = 15,
  height = 14
)

Arguments

gene       gene symbol, case sensitive
dta        synthetic lethal RNAi screen data
controlsiRNA controlsiRNA could be a vector of several siRNA, including positive/negative control
FILEPATH   path to store the figure
colour      colour used in graphs
zPrimeMed   zPrime factor basing on median
zPrimeMean  zPrime factor basing on mean
treatment   the treatment condition in EXPERIMENT_MODIFICATION
tTest

control the control condition in EXPERIMENT_MODIFICATION

normMethod could be a PLATE and negative controls

save_plot whether save a png file in the working directory.

width width of the plot

height height of the plot

**Value**

Return the ggplot2 objects in a list, which could be plotted individually.

**Examples**

```r
data(example_dt)
zF_mean <- zFactor(example_dt, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1")
zF_med <- zFactor(example_dt, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1", useMean = FALSE)
p01 <- siRNAPlot("AAK1", example_dt, controlsRNA = c("lipid only", "scrambled control si1"), FILEPATH = ".", zPrimeMed = zF_med, zPrimeMean = zF_mean, treatment = "treatment", control = "control", normMethod = c("PLATE", "lipid only", "scrambled control si1"))
```

---

**tTest**

_Student’s t-test on B-score_

**Description**

Select hits by student’s t-test using B-score from treatment and control plates.

**Usage**

```r
tTest(mtx, n_treat, n_cont)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mtx</td>
<td>b-score matrix.</td>
</tr>
<tr>
<td>n_treat</td>
<td>number of treatment plates</td>
</tr>
<tr>
<td>n_cont</td>
<td>number of control plates</td>
</tr>
</tbody>
</table>

**Value**

A list containing student’s t-test for each master plate

- **pvalue**: p-value of the t-test
- **Treat_Cont**: difference in bscore: treatment - control
- **p_adj**: BH adjusted p-value
zFactor

References

Examples
data(example_dt)
bscore_res <- sapply(unique(example_dt$MASTER_PLATE), bScore,
        example_dt, control = "control", treatment = "treatment", simplify = FALSE)
tTest(bscore_res$P001, 3, 3)

zFactor

Calcualte the Z and Z' factor

Description
calcualte the Z and Z' factor for each plate.

Usage
zFactor(dta, negativeCon, positiveCon, useMean = TRUE)

Arguments
dta  synthetic lethal RNAi screen data.
negativeCon  the negative control used in the WELL_CONTENT_NAME.
positiveCon  the positive control used in the WELL_CONTENT_NAME.
useMean  use mean to calcualate z factor and z' factor by default; otherwise use median.

Value
A data.frame contains z factor and z' factor

References

Examples
data(example_dt)
res <- zFactor(example_dt, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1")
Index

* datasets
  example_dt, 3

bScore, 2
example_dt, 3
madSelect, 4
plateHeatmap, 5
rankProdHits, 6
rsaHits, 7
scatterPlot, 8
siRNAPlot, 9
tTest, 10
zFactor, 11