Package ‘synlet’

March 29, 2017

Type Package
Title Hits Selection for Synthetic Lethal RNAi Screen Data
Version 1.4.0
Date 2015-07-31
Author Chunxuan Shao <c.shao@dkfz.de>
Maintainer Chunxuan Shao <c.shao@dkfz.de>
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
LazyData TRUE
biocViews CellBasedAssays, QualityControl, Preprocessing, Visualization, FeatureExtraction
Depends R (&gt;= 3.2.0), ggplot2
Imports doBy, dplyr, grid, magrittr, RColorBrewer, RankProd, reshape2
Suggests knitr, testthat
VignetteBuilder knitr
NeedsCompilation no

R topics documented:

bScore .................................................. 2
exampleDat ............................................. 2
madSelect .............................................. 3
plateHeatmap ........................................... 4
rankProdHits ........................................... 4
rsahits .................................................. 5
scatterPlot ............................................. 7
siRNAPlot .............................................. 7
tTest .................................................... 8
zFactor .................................................. 9
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**

\[ \text{bScore(masterPlate, dat, treatment, control, outFile = FALSE)} \]

**Arguments**

- `masterPlate`: a master plate to be normalized
- `dat`: 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
bscore.res <- sapply(as.character(unique(exampleDat$MASTER_PLATE)), bScore, exampleDat, control = "control", treatment = "treatment", simplify = FALSE)
```

---

*exampleDat*

**Synthetic lethal RNAi screen example data.**

**Description**

A dataset containing synthetic lethal RNAi screen data to show how functions work. The variables are as follows:

**Usage**

```r
data(exampleDat)
```
Format
A data frame 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 frame containing RANi screen data, the READOUT value has no real biological meaning.

\[
\text{madSelect} \quad \text{Select hits basing on median + k*MAD}
\]

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

Usage
\[
\text{madSelect} \left( \text{masterPlate}, \text{dat}, k = 3, \text{treatment}, \text{control}, \text{outFile} = \text{FALSE}, \text{normMethod} = \text{"PLATE"} \right)
\]

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?
rankProdHits

References

Examples
madSelection <- sapply(as.character(unique(exampleDat$MASTER_PLATE)),
madSelect, exampleDat, control = "control",
treatment = "treatment", simplify = FALSE)
madSelection.c <- do.call(rbind,
  lapply(names(madSelection), function(x) madSelection[[x]]))

plateHeatmap
Heatmap of all plates

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

Usage
plateHeatmap(dat, baseSize = 12)

Arguments
dat synethetic lethal RNAi screen data
baseSize basic font size used for x/y axis and title for heatmaps

Value
a ggplot object

Examples
tem.1 <- plateHeatmap(exampleDat)
ggsave("platesHeatmap.pdf", plot = tem.1, width = 500, height = 500, limitsize = FALSE)

rankProdHits
Select hits by the rank product method

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

Usage
rankProdHits(masterPlate, dat, treatment, control, normMethod = "PLATE")
**rsaHits**

**Arguments**

- **masterPlate**: the master plate to be analyzed
- **dat**: 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
rankp.res <- sapply(as.character(unique(exampleDat$MASTER_PLATE)),
                    rankProdHits, exampleDat, control = "control",
                    treatment = "treatment", simplify = FALSE)
rankp.c <- data.frame(do.call(rbind, lapply(names(rankp.res), function(x) rankp.res[[x]]))))
```

**rsaHits**

*Select hits by RSA*

**Description**

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

**Usage**

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

- **dat**: 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

```
rsaHits(exampleDat, 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

scatterPlot(dat, controlOnly = FALSE, colour, ...)

Arguments

dat          synthetic lethal RNAi screen data
controlOnly  whether or not to plot control wells only
colour       colour for different signals
...          positive/negative signals, must be specified

Value

a ggplot object

Examples

scatterPlot(exampleDat, controlOnly = FALSE, colour = rainbow(10),
"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, dat, controlsiRNA, FILEPATH = ".", colour = rainbow(10),
zPrimeMed, zPrimeMean, treatment, control, normMethod = c("PLATE"),
width = 15, height = 14)
**tTest**

**Arguments**

- `gene`: gene symbol, case sensitive
- `dat`: 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
- `control`: the control condition in EXPERIMENT_MODIFICATION
- `normMethod`: could be a PLATE and negative controls
- `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
ezF_mean <- zFactor(exampleDat, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1")
zF_med <- zFactor(exampleDat, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1", useMean = FALSE)
tem.1 <- siRNAPlot("AAK1", exampleDat, controlsiRNA = 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"))
```

**Description**

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

**Usage**

`tTest(masterPlate, bScore, numTreat, numCont)`

**Arguments**

- `masterPlate`: the master plate to be analyzed
- `bScore`: normalized bScore
- `numTreat`: number of treatment plates
- `numCont`: number of control plates
zFactor

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

References


Examples

```r
bscore.res <- sapply(as.character(unique(exampleDat$MASTER_PLATE)), bScore,
exampleDat, control = "control", treatment = "treatment", simplify = FALSE)
bscore.ttest <- sapply(names(bscore.res), tTest, bscore.res, numTreat = 3,
numCont = 3, simplify = FALSE, USE.NAMES = TRUE)
bscore.combined <- data.frame(do.call(rbind, lapply(names(bscore.ttest),
function(x) if (!is.null(bscore.ttest[[x]])) (data.frame(MASTER_PLATE = x,
siRNAs = rownames(bscore.ttest[[x]]), bscore.ttest[[x]])))))
```

---

zFactor  
Calulate the Z and Z' factor

Description

calculate the Z and Z' factor for each plate.

Usage

`zFactor(dat, negativeCon, positiveCon, useMean = TRUE)`

Arguments

dat  
- 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 calculate z factor and z' factor by default; otherwise use median.

Value

A data.frame contains z factor and z' factor

References

Examples

zFactor(exampleDat, negativeCon = "scrambled control si1", positiveCon = "PLK1 si1")
Index

«Topic datasets
  exampleDat, 2
  bScore, 2
  exampleDat, 2
  madSelect, 3
  plateHeatmap, 4
  rankProdHits, 4
  rsaHits, 5
  scatterPlot, 7
  siRNAPlot, 7
  tTest, 8
  zFactor, 9