Package ‘RNAither’

April 26, 2017

Title  Statistical analysis of high-throughput RNAi screens

Version 2.24.0

Author  Nora Rieber and Lars Kaderali, University of Heidelberg,
         Viroquant Research Group Modeling, Im Neuenheimer Feld 267,
         69120 Heidelberg, Germany

Maintainer Lars Kaderali <lars.kaderali@uni-greifswald.de>

Depends  R (>= 2.10), topGO, RankProd, prada

Imports  geneplotter, limma, biomaRt, car, splots, methods

Description  RNAither analyzes cell-based RNAi screens, and includes
             quality assessment, customizable normalization and statistical
             tests, leading to lists of significant genes and biological
             processes.

License  Artistic-2.0

biocViews  CellBasedAssays, QualityControl, Preprocessing,
            Visualization, Annotation, GO

NeedsCompilation  no

R topics documented:

RNAither-package .................................................. 3
BScore ................................................................. 4
channelPlot .......................................................... 5
closestToZero ....................................................... 6
compareHits .......................................................... 6
compareReplicaPlates .............................................. 7
compareReplicates ................................................ 8
compareReplicateSD ............................................... 9
compareReplicateSDPerScreen .................................. 10
controlDensity ..................................................... 11
controlDensityPerPlate ......................................... 12
controlDensityPerScreen ....................................... 13
controlNorm ......................................................... 14
createSubset ....................................................... 15
dataset ............................................................. 15
datasetDrosophila ................................................ 16
discardLabtek .................................................... 16
discardWells ....................................................... 17
divideChannels .................................................. 17
divNorm ............................................................. 18
DRQualControl .................................................... 19
eraseDataSetColumn ............................................. 20
findReplicates ..................................................... 20
furthestFromZero ................................................ 21
generateDatasetFile ............................................. 21
generateReplicateMat .......................................... 24
generateRepMatNoFilter ....................................... 25
gseaAnalysis ....................................................... 26
header ............................................................... 27
headerDrosophila ............................................... 27
hitselectionPval ................................................ 27
hitselectionZscore .............................................. 29
hitselectionZscorePval .......................................... 30
incorporatepValVec ............................................ 32
indexSubset ....................................................... 33
joinDatasetFiles ................................................. 33
joinDatasets ...................................................... 34
LiWongRank ....................................................... 35
lowessNorm ....................................................... 36
mainAnalysis ..................................................... 37
makeBoxplot4PlateType ......................................... 38
makeBoxplotControls .......................................... 39
makeBoxplotControlsPerPlate ................................ 40
makeBoxplotControlsPerScreen ................................ 41
makeBoxplotPerPlate .......................................... 42
makeBoxplotPerScreen ......................................... 43
MannWhitney ..................................................... 44
multiTestAdjust ................................................ 45
numCellQualControl .......................................... 45
orderGeneIDs .................................................... 46
percCellQualControl .......................................... 47
plotBar ............................................................. 48
plotControlHisto ............................................... 49
plotControlHistoPerplate ..................................... 50
plotControlHistoPerscreen .................................... 51
plotHisto ........................................................ 52
plotHistoPerplate .............................................. 52
plotHistoPerscreen ............................................. 53
plotQQ ............................................................. 54
plotQQperplate ................................................ 55
plotQQperscreen ................................................. 56
pValVec1 .......................................................... 57
pValVec2 .......................................................... 57
quantileNormalization ......................................... 58
RankProduct ...................................................... 59
replicatesCV ..................................................... 60
replicatesSpearmancor ........................................ 61
rms ................................................................. 62
rmaither .......................................................... 62
saveDataset ....................................................... 65
RNAither-package

Description

RNAither analyzes cell-based RNAi screens, and includes quality assessment, customizable normalization and statistical tests, leading to lists of significant genes and biological processes.

Details

Package: RNAither
Type: Package
Version: 1.0
Date: 2008-07-20
License: Artistic License 2.0

Author(s)

Nora Rieber and Lars Kaderali

BScore

**BScore normalization**

**Description**
Normalization with BScores (see References).

**Usage**

BScore(header, dataset, listOfArgs)

**Arguments**

- **header**
  the header of a dataset file generated with `generateDatasetFile`

- **dataset**
  an R data frame generated with `generateDatasetFile`

- **listOfArgs**
  a list containing:
  - a character string specifying the column whose values will be used for normalization
  - a flag specifying whether controls should be excluded for the computation of the median polish (1) or not (0)

**Value**

A list containing:

- **header**
  The new header (with an added entry about the normalization procedure in the comments)

- **dataset**
  The new dataset with normalized values. The old values are saved in an extra column of the dataset with the suffix ".old"

**References**


**Examples**

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normres <- BScore(header, dataset, list("SigIntensity", 0))
newheader <- normres[[1]]
newdataset <- normres[[2]]
channelPlot

Plot signal channels against each other

Description
Generates plots allowing pairwise comparison of signal channels. Fits a lowess regression curve into the plots.

Usage
channelPlot(header, dataset, vecOfChannels, flag, plotTitle, showPlot, smSpan=2/3)

Arguments
- header: the header of a dataset file generated with `generateDatasetFile`
- dataset: an R data frame generated with `generateDatasetFile`
- vecOfChannels: A vector containing the names of the signal channels to be compared, e.g. "Sig-Intensity"
- flag: 0, 1, or 2. 0 uses the data from the complete dataset, 1 makes comparisons for each experiment, 2 makes comparisons for each plate.
- plotTitle: The plot title
- showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.
- smSpan: The smoother span of the lowess curve. This gives the proportion of points in the plot which influence the smooth at each value. Larger values give more smoothness. Optional, defaults to 2/3

Value
Saves the plots in pdf and png files named after the experiment name specified in the header concatenated with the plotTitle, the number of the comparison, and if applicable the experiment number and/or the plate number.
When flag == 0, returns the plot name (plotName).
When flag == 1, returns a list containing:
- plotName: The plot name
- minOfScreens: The number of the first experiment
- numOfScreens: The number of the last experiment
When flag == 2, returns a list containing: the plot name, a vector with the number of the first experiment and of the last experiment, and a vector with the number of the first plate and the number of the last plate.

Examples
```
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
plotname <- channelPlot(header, dataset, c("SigIntensity", "NbCells"), 0, "Channel comparison", 1)
```
**closestToZero**  
*Return the replicate value closest to zero*

**Description**  
Out of a set of replicate values, returns the one closest to zero.

**Usage**  

```r  
closestToZero(Ivec, na.rm = T)  
```

**Arguments**

- `Ivec`: All channel values for a specific siRNA/gene
- `na.rm`: Removes NA values

**Value**  

A double giving the value closest to zero out of the given replicate values.

**See Also**

- `rms`, `trim`, `furthestFromZero`, `summarizeReps`, `summarizeRepsNoFiltering`

**Examples**

```r  
data(exampleDataset, package="RNAither")  
Indexes <- findReplicates(dataset, "GeneName", "CPSF1")  
replicateclosest <- closestToZero(dataset$SigIntensity[Indexes])  
```

---

**compareHits**  
*Searching for common hits between different scoring methods*

**Description**  
Searches for common hits between different scoring methods.

**Usage**  

```r  
compareHits(hitVec1, hitVec2, namesHitVec1, namesHitVec2)  
```

**Arguments**

- `hitVec1, hitVec2`: the two binary hit vectors to be compared
- `namesHitVec1, namesHitVec2`: the names of the siRNAs corresponding to the hit vectors
compareReplicaPlates

Compare replica plates

Description
Generates plots comparing the same plates in different experiments pairwise.

Usage
compareReplicaPlates(header, dataset, plotTitle, col4val, showPlot)

Arguments

- header: the header of a dataset file generated with generateDatasetFile
- dataset: an R data frame generated with generateDatasetFile
- plotTitle: the plot title
- col4val: a character string specifying the column whose values will be used for the plot
- showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

Value
For each plate, plots of pairwise comparisons between replicate intensities are generated and saved as a pdf file named after the experiment name specified in the header concatenated with the plotTitle.
compareReplicates

See Also

compareReplicates

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

compareReplicaPlates(header, dataset, "Comparison of replica plate", "SigIntensity", 1)

compareReplicates Compare replicate values

Description

Plots replicate intensities pairwise for each experiment.

Usage

compareReplicates(header, dataset, plotTitle, col4val, col4anno, plotDesign, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
plotTitle the plot title
col4val a character string specifying the column whose values will be used for the plot
col4anno a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

Value

For each experiment, plots of pairwise comparisons between replicate intensities are generated and saved as a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle, and the number of the experiment.

The function returns a list containing:

plotName the plot name
minOfScreens the number of the first experiment
numOfScreens the number of the last experiment
maxCombinationNum the number of replicates to compare

See Also

compareReplicaPlates
Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

compareReplicates(header, dataset, "Comparison of Replicates", "SigIntensity", "GeneName", 1, 0)

---

**compareReplicateSD**

Plot the standard deviation of replicates

Description

In the same fashion as `spatialDistrib`, generates a plot of the standard deviation of replicate values.

Usage

```
compareReplicateSD(header, dataset, plotTitle, colname4SD, col4anno, showPlot)
```

Arguments

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `plotTitle`: the plot title
- `colname4SD`: a character string specifying the column whose values will be used for the computation of the replicate standard deviation
- `col4anno`: a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

Value

Generates a plot of the standard deviation of replicate values of all experiments. The plot is saved as a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

Wells showing positive controls sd are marked with a "P", wells showing negative controls sd with an "N".

The plot will also be saved as an html file containing mouse-overs with the siRNA name for each well.

The function returns the plotname.

See Also

`spatialDistrib`, `compareReplicateSDPerScreen`
compareReplicateSDPerScreen

Plot the standard deviation of replicates for each experiment

Description

In the same fashion as spatialDistrib, generates plots of the standard deviation of replicate values for each experiment.

Usage

compareReplicateSDPerScreen(header, dataset, plotTitle, colname4SD, col4anno, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
plotTitle the plot title
colname4SD a character string specifying the column whose values will be used for the computation of the replicate standard deviation
col4anno a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

Value

Generates plots of the standard deviation of replicate values for each experiment. The plots are saved as png files named after the experiment name specified in the header concatenated with the plotTitle and the number of the experiment.

Wells showing positive controls sd are marked with a "P", wells showing negative controls sd with an "N".

The plots will also be saved as html files containing mouse-overs with the siRNA name for each well.

The function returns a list of length 3 containing:

basicPlotName the plot name
minOfScreens the number of the first experiment
numOfScreens the number of the last experiment

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

compareReplicateSD(header, dataset, "Replicate standard intensity deviation", "SigIntensity", "GeneName", 1)
controlDensity

See Also

controlDensityPerScreen, controlDensityPerPlate

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

calculateReplicateSDDensityPerScreen(header, dataset, "Replicate standard intensity deviation", "SigIntensity", "GeneName", 1)
controlDensityPerPlate

Plotting the control density per plate

Description
Plots the density of positive and negative controls (if applicable) for each plate.

Usage
controlDensityPerPlate(header, dataset, channel, plotTitle, plotDesign, showPlot, supHisto)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the name of the column containing the values for computing the density, e.g. "SigIntensity"
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows
supHisto 0 or 1. 1 will additionally superimpose a colour histogram of the values for the positive and negative controls. Otherwise choose 0.

Value
Generates a series of plots for each experiment and each plate, showing the density of positive and negative controls (if applicable). The plots are saved as pdf and png files named after the experiment name specified in the header concatenated with the plotTitle.

The function returns a list of length 3 containing:

plotName the plot name
c(minOfScreens, numOfScreens) a vector with the number of the first experiment and of the last experiment
c(minOfPlates, numOfPlates) a vector with the number of the first plate and the number of the last plate

See Also
ccontrolDensity, controlDensityPerScreen

Examples
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

controlDensityPerPlate(header, dataset, "SigIntensity", "Control density", 1, 1, 1)
controlDensityPerScreen

Plotting the control density per experiment

Description
Plots the density of positive and negative controls (if applicable) for each experiment.

Usage
controlDensityPerScreen(header, dataset, channel, plotTitle, showPlot, supHisto)

Arguments
- header: the header of a dataset file generated with `generateDatasetFile`
- dataset: an R data frame generated with `generateDatasetFile`
- channel: a character string specifying the name of the column containing the values for computing the density, e.g., "SigIntensity"
- plotTitle: the plot title
- showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows
- supHisto: 0 or 1. 1 will additionally superimpose a colour histogram of the values for the positive and negative controls. Otherwise choose 0.

Value
Generates a series of plots for each experiment, showing the density of positive and negative controls (if applicable). The plots are saved as pdf and png files named after the experiment name specified in the header concatenated with the `plotTitle` and the number of the experiment.

The function returns a list of length 3 containing:
- `plotName`: the plotname
- `minOfScreens`: the number of the first experiment
- `numOfScreens`: the number of the last experiment

See Also
- `controlDensity`
- `controlDensityPerPlate`

Examples
```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
controlDensityPerScreen(header, dataset, "SigIntensity", "Control density", 1, 1)
```
controlNorm

Normalization on controls

Description

Performs a normalization on either positive or negative controls.

Usage

controlNorm(header, dataset, listOfArgs)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
listOfArgs a list containing:
- a character string specifying the column whose values will be used for normalization
- 1 or 2, 1 meaning a normalization per experiment, 2 meaning a normalization per plate
- 0 or 1, 0 meaning a normalization on the median of negative controls, 1 meaning a normalization on the median of positive controls. Can also be the GeneName of a specific control siRNA
- 1 or 2, 1 meaning the signal values are divided by the median, 2 meaning the median is subtracted from the signal values

Value

Returns a list containing:

header the new header (with an added entry about the normalization procedure in the comments).
dataset the new dataset with normalized values. The old values are saved in an extra column with the suffix ".old".

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- controlNorm(header, dataset, list(2, 0, "SigIntensity", 1))

newheader <- normres[[1]]
newdataset <- normres[[2]]
**createSubset**  

Creating a subset of a dataset according to a certain column value

**Description**

Creates a subset of a dataset containing all wells/lines having a certain value in a specified column.

**Usage**

```r
createSubset(dataset, listIDs, equalTo)
```

**Arguments**

- `dataset` an R data frame generated with `generateDatasetFile`
- `listIDs` a character string and one of the following: `Spotnumber`, `Internal_GeneID`, `GeneName`, `SpotType`, `SigIntensity`, `SDSIntensity`, `Background`, `LabtekNb`, `RowNb`, `ColNb`, `ScreenNb`, `NbCells`, `PercCells`, ...
- `equalTo` A value or character string specifying the value in the chosen column, e.g. all wells on plate 2

**Value**

A subset of the dataset containing only the wells/lines having a certain value in a specified column.

**See Also**

`indexSubset`

**Examples**

```r
data(exampleDataset, package="RNAither")
subset <- createSubset(dataset, dataset$LabtekNb, 2)
```

---

**dataset**

*a typical example RNAi dataset*

**Description**

See `generateDatasetFile` for details

**Usage**

```r
dataset
```

**Format**

See `generateDatasetFile`
datasetDrosophila  

*Genome-wide RNAi screen of cell viability in Drosophila Kc167 cells by Boutros et al.*

**Description**


**Usage**

```
datasetDrosophila
```

**Format**

see `generateDatasetFile` for details

---

discardLabtek  

*Remove a complete plate from the analysis*

**Description**

Removes a plate/LabTek from the analysis by setting its spot type in the dataset to -1.

**Usage**

```
discardLabtek(data, screenNr, labtekNr)
```

**Arguments**

- `data` an R data frame generated with `generateDatasetFile`
- `screenNr` the number of the experiment that contains the plate to discard
- `labtekNr` the number of the plate to discard

**Value**

A new dataset that still contains the specified plate/LabTek, but excludes it from the further analysis by setting its SpotTypes to -1.

**See Also**

`discardWells`

**Examples**

```
data(exampleDataset, package="RNAither")
newdataset <- discardLabtek(dataset, 2, 2)
```
**discardWells**

Remove wells from the analysis

**Description**

Removes wells from the analysis by setting their spot type in the dataset to -1.

**Usage**

```r
discardWells(data, screenNr, labtekNr, vecPositions)
```

**Arguments**

- `data`: an R data frame generated with `generateDatasetFile`
- `screenNr`: the number of the experiment that contains the plate to discard
- `labtekNr`: the number of the plate to discard
- `vecPositions`: a vector specifying the numbers of the wells to discard

**Value**

A new dataset that does not contains the specified wells. A new dataset that still contains the specified wells/spots, but excludes them from the further analysis by setting their SpotTypes to -1.

**See Also**

- `discardLabtek`

**Examples**

```r
data(exampleDataset, package="RNAither")
newdataset <- discardWells(dataset, 2, 1, c(1, 10, 15))
```

---

**divideChannels**

Divide channel values

**Description**

Replace two channels by their ratio.

**Usage**

```r
divideChannels(ch1, ch2)
```

**Arguments**

- `ch1`: a vector giving all values from channel 1
- `ch2`: a vector giving all values from channel 2
divNorm

Value

A vector of the ratio of channel 1 and channel 2.

See Also

sumChannels

Examples

data(exampleDataset, package="RNAither")

newch <- divideChannels(dataset$SigIntensity, dataset$NbCells)

Description

Normalization with the mean, median, or any other function.

Usage

divNorm(header, dataset, listOfArgs)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
listOfArgs a list containing:
- a character string specifying the column whose values will be used for normalization
- a function to be used for the normalization, e.g. mean, median, ...
- 1 or 2, 1 meaning a normalization per experiment, 2 meaning a normalization per plate
- 1 or 2, 1 meaning the normalization is achieved by a division of the intensity values by the outcome of funname, 2, meaning by a substraction
- a flag specifying whether controls should be excluded for the computation of the result of the function specified in the first element (1) or not (0).

Value

Returns a list containing:

header the new header (with an added entry about the normalization procedure in the comments)
dataset the new dataset with normalized values. The old values are saved in an extra column with the suffix ".old"
**Example**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- divNorm(header, dataset, list(median, 2, 1, "SigIntensity", 1))

newheader <- normres[[1]]
newdataset <- normres[[2]]
```

---

**Description**

Computes the dynamic range per plate for a complete dataset file and plots the results.

**Usage**

```r
DRQualControl(header, data, nbLinesHeader, channel, plotTitle, showPlot)
```

**Arguments**

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `data`: an R data frame generated with `generateDatasetFile`
- `nbLinesHeader`: typically 3
- `channel`: A character string specifying the name of the column containing the values for computing the dynamic range, e.g. "SigIntensity"
- `plotTitle`: the plot title
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

**Value**

Returns the dynamic range for each plate in the shell and saves them in a text file. The name of the text file will be the concatenation of the experiment name specified in the header and the character string "DR.txt".

Shows a plot of the dynamic range values and saves it as a pdf file under the experiment name specified in the header concatenated with the function argument `plotTitle`.

**References**


**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

DRQualControl(header, dataset, 3, "SigIntensity", "DR per plate", 1)
```
eraseDataSetColumn  Remove columns from dataset

Description
Removes a specified column from a dataset.

Usage
eraseDataSetColumn(dataset, colname)

Arguments
- `dataset`: an R data frame generated with `generateDatasetFile`
- `colname`: a character string specifying the name of the column to be removed

Value
An R data frame with the specified column removed.

Examples
```r
data(exampleDataset, package="RNAither")
newdataset <- eraseDataSetColumn(dataset, "SDSIntensity")
```

findReplicates  Find all replicates of a certain siRNA/gene in a dataset

Description
Returns which lines in the dataset correspond to a given siRNA/gene ID.

Usage
findReplicates(dataset, whichCol, replicateID)

Arguments
- `dataset`: an R data frame generated with `generateDatasetFile`
- `whichCol`: a character string specifying the name of the column containing the ID, either `Internal_GeneID` or `GeneName`
- `replicateID`: the siRNA/gene ID of interest

Value
An integer vector containing the indexes in the main dataset of all wells corresponding to a given siRNA/gene ID
furthestFromZero

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

Indexes <- findReplicates(dataset, "GeneName", "CPSF1")

furthestFromZero

Return the replicate value furthest from zero

Description

Out of a set of replicate values, returns the one furthest from zero.

Usage

furthestFromZero(Ivec, na.rm = T)

Arguments

Ivec All channel values for a specific siRNA/gene
na.rm Removes NA values

Value

A double giving the value furthest from zero out of the given replicate values.

See Also

rms, trim, closestToZero, summarizeReps, summarizeRepsNoFiltering

Examples

data(exampleDataset, package="RNAither")

Indexes <- findReplicates(dataset, "GeneName", "CPSF1")
replicateclosest <- furthestFromZero(dataset$SigIntensity[Indexes])

generateDatasetFile

Generate Dataset File

Description

Generates a text file containing all experimental data. Needed for all subsequent analysis functions.

Usage

generateDatasetFile(externalExperimentName, typeOfData, comments, outputFile, plateLayoutInternal, plateLayoutNCBI, nbRowsPerPlate, nbColsPerPlate, screenNb_pre, emptyWells, poorWells, controlCoordsOutput, backgroundValOutput, meanSignalOutput, SDmeanSignal, objNumOutput, cellNumOutput)
Arguments

- **externalExperimentName**: A character string specifying the experiment name, e.g. "Johns Experiment Nb. 1"
- **typeOfData**: A character string specifying the type of data, e.g. "364 well plate data for virus screens"
- **comments**: A character string specifying comments. NA if not available.
- **outputFile**: A character string specifying the name of the text file containing the dataset.
- **plateLayoutInternal**: A matrix of internal siRNA IDs specifying their position on the plate (row-wise). Each column of the matrix stands for one plate.
- **plateLayoutNCBI**: A matrix of gene names specifying their position on the plate (row-wise). Each column of the matrix stands for one plate.
- **nbRowsPerPlate**: The number of rows per plate
- **nbColsPerPlate**: The number of columns per plate
- **screenNb_pre**: The screen/experiment number
- **emptyWells**: A list containing, for each plate, an integer vector of the positions of empty wells. NA if there are no empty wells on the plate.
- **poorWells**: A list containing, for each plate, an integer vector of the positions of wells that, for a certain reason, should not be taken into account during the analysis. NA if there are no such wells on the plate.
- **controlCoordsOutput**: A list containing, for each plate, a list of integer vectors specifying the positions of positive (first element in sublist) and negative (second element in sublist) controls. NA if there are no positive/negative controls on the plate.
- **backgroundValOutput**: A list containing, for each plate, a vector of background values per well
- **meanSignalOutput**: A list containing, for each plate, a vector of intensity values for each well
- **SDmeanSignal**: A list containing, for each plate, a vector of standard deviations of intensity values for each well
- **objNumOutput**: A list containing, for each plate, a vector of the number of identified objects for each well
- **cellNumOutput**: A list containing, for each plate, a vector of intensity values for each well, e.g. a vector of the number of identified cells for each well.

Details

Positions on plates are specified with one integer only. For example, the position of the well in row 2 and column 5 is \((\text{RowNo}-1)\times\text{(Number of columns on plate)}+\text{ColNo}\).

Value

The function generates a text file consisting of a header and a 'dataset'. The header contains the experiment description (`externalExperimentName`, `typeOfData` and `comments`). The dataset is an R data frame, each row corresponding to one well, with the following columns:

- **Spotnumber**: The position of the well on the plate
Internal_GeneID
The ID of the siRNA

GeneName
The gene name

SpotType
Can be -1, 0, 1 or 2.
Type -1 wells (e.g. empty wells, wells with poor quality) are not considered in subsequent analyses but are kept in the data set for the sake of completeness.
Type 0 wells correspond to negative controls, type 1 wells to positive controls.
Type 2 wells correspond to the standard data wells.

SigIntensity
The signal intensity (channel 1)

SDSIntensity
The standard deviation of the signal intensity, if available

Background
The background per well, if available

LabtekNb
The plate number

RowNb
The row number

ColNb
The column number

ScreenNb
The screen number

NbCells
E.g. the number of cells identified in the well (channel 2)

PercCells
The ratio (number of identified cells)/(number of identified objects)

See Also
joinDatasetFiles, joinDatasets

Examples

##gene names
plateLayout1 <- c("test1", "empty", "test3", "test4", "test5",
"test6", "test7", "empty", "test9", "test10", "test11", "test12")

plateLayout2 <- c("test1", "test2", "test3", "test4", "test5",
"test6", "test7", "test8", "test9", "test10", "test11", "test12")

plateLayout <- cbind(plateLayout1, plateLayout2)

emptyWells <- list(c(2, 8), NA_integer_)
##the first plate has two empty wells at position 2 and 8,
##the second plate does not have any empty wells

poorWells <- NA_integer_  
##no wells of poor quality

controlCoordsOutput <- list(list(NA_integer_, NA_integer_), list(NA_integer_, c(9,10)))
##the first plate does not have any control siRNAs,
##the second plate has two negative controls at position 9 and 10

backgroundValOutput<-NA_integer_
##no background signal intensities available

sigPlate1<-c(2578, NA_integer_, 3784, 3784, 2578, 5555, 5555, NA_integer_, 8154, 2578, 3784, 2578)
sigPlate2<-c(8154, 3784, 5555, 3784, 11969, 2578, 1196, 5555, 17568, 2578, 5555, 2578)
##the signal intensities on the plates
meanSignalOutput<-list(sigPlate1, sigPlate2)

SDmeansignal<-NA_integer_
##no standard deviation available

objnumOutput<-NA_integer_
##no cell count available

cellnumOutput<-NA_integer_

generateDatasetFile("First test screen", "RNAi in virus-infected cells", NA_character_, "testscreen_output.txt", plateLayout, plateLayout, 3, 4, 1, emptyWells, poorWells, controlCoordsOutput, backgroundValOutput, meanSignalOutput, SDmeansignal, objnumOutput, cellnumOutput)

##load the dataset into R:
header<-readLines("testscreen_output.txt",3)

generateReplicateMat

Generate a matrix of replicates

Description

Generates a matrix out of a dataset, each row corresponding to an siRNA/gene ID, each column to a channel value or its index in the dataset.

Usage

generateReplicateMat(data, minNbReps, IndexOrInt, col4val, col4anno)

Arguments

data an R data frame generated with generateDatasetFile

minNbReps set to 2 if you want to exclude replicates occurring only once in the dataset, otherwise 1.

IndexOrInt a character string - either "Index" or "Intensities" - specifying which values are to be contained in the output matrix.

col4val a character string specifying the name of the dataset column to be used for the values of the output matrix (if IndexOrIntensities is set to "Intensities"), for example "SigIntensity" or "NbCells"

col4anno a character string specifying the name of the dataset column to be used for the output matrix' rows, for example "GeneName" or "Internal_GeneID".

Details

The function will omit values or indexes of lines/wells whose value in the column specified by colname4val is set to NA, (which is the case if the spot type is set to -1). If you do not want to omit those, use generateRepMatNoFilter.
**Value**

A matrix with each row corresponding to an siRNA/gene ID (as reflected in rownames), each column to a channel value or its index in the dataset. Missing values (in case of different number of replicates occurring for different siRNAs/genes) are set to NA.

**See Also**

`generateRepMatNoFilter`

**Examples**

```r
data(exampleDataset, package="RNAither")
replicatematrix <- generateReplicateMat(dataset, 2, "Index", "SigIntensity", "GeneName")
```

---

**generateRepMatNoFilter**

*Generate a matrix of replicates (II)*

**Description**

Generates a matrix out of a dataset, each row corresponding to an siRNA/gene ID, each column to a channel value or its index in the dataset.

**Usage**

`generateRepMatNoFilter(data, minNbReps, IndexOrInt, col4val, col4anno)`

**Arguments**

- `data`: an R data frame generated with `generateDatasetFile`
- `minNbReps`: set to 2 if you want to exclude replicates occurring only once in the dataset, otherwise 1.
- `IndexOrInt`: a character string - either "Index" or "Intensities" - specifying which values are to be contained in the output matrix.
- `col4val`: a character string specifying the name of the dataset column to be used for the values of the output matrix (if `IndexOrInt` is set to "Intensities"), for example "SigIntensity" or "NbCells".
- `col4anno`: a character string specifying the name of the dataset column to be used for the output matrix' rows, for example "GeneName" or "Internal_GeneID".

**Details**

The function will not omit values or indexes of lines/wells with spot type -1. If you want to omit those, use `generateReplicatematrix`.

**Value**

A matrix with each row corresponding to an siRNA/gene ID (as reflected in rownames), each column to a channel value or its index in the dataset. Missing values (in case of different number of replicates occurring for different siRNAs/genes) are set to NA.
gseaAnalysis

Perform a GSEA analysis of a list of genes

Description
Performs a GSEA analysis of a list of genes using the package `topGO` (see References).

Usage
\[
gseaAnalysis(hitVector, whichOnto)
\]

Arguments
- `hitVector`: a named hit vector as generated by `hitselectionZscore` or `hitselectionPval`
- `whichOnto`: One of the three GO ontologies: "biological_process", "molecular_function" or "cellular_component"

Value
A table containing the enriched GO terms and their significance.

References

See Also
Ttest

Examples
\[
data(scoredDataset1, package="RNAither")
data(pValVec1, package="RNAither")

##for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

scoredHits1 <- hitselectionPval(scoredDataset1, pValVec1, "SigIntensity", "Hits1", 0.1, "GeneName", "pvalue_testfile1.txt")
hitVector1 <- scoredHits1[[2]]
gseaTable <- gseaAnalysis(hitVector1, "biological_process")
\]
**header**  
a typical header of an example RNAi dataset

**Description**  
See `generateDatasetFile` for details

**Usage**  
header

**Format**  
See `generateDatasetFile`

---

**headerDrosophila**  
the header of the genome-wide RNAi screen of cell viability in *Drosophila Kc167* cells by Boutros et al.

**Description**  
M. Boutros et al., Genome-wide RNAi analysis of growth and viability in Drosophila cells, Science, 303(5659):832-835, 2004. 3, 18

**Usage**  
headerDrosophila

**Format**  
See `generateDatasetFile`

---

**hitselectionPval**  
Selecting hits according to p-values

**Description**  
Selects significant genes according to their p-value.

**Usage**  
`hitselectionPval(dataset, pValVec, col4val, col4sel, thresh, col4anno, file4hits)`
hitselectionPval

Arguments

- **dataset**: an R data frame generated with `generateDatasetFile`
- **pValVec**: a vector of p-values, as generated by one of the test functions `Ttest`, `MannWhitney` or `RankProduct`
- **col4val**: a character vector specifying a column of intensity values
- **col4sel**: a character vector specifying the name of the new dataset column where hits will be stored
- **thresh**: the threshold for the p-values, typically 0.05
- **col4anno**: a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
- **file4hits**: the name of the file to store the results in

Details

If there are no p-values under the defined threshold `thresh`, the threshold is increased to `min(pvalvec)`.

Value

A list containing:

- **dataset**: the dataset with an added column defining the hits in the form of a binary vector
- **hitVector**: the binary vector itself
- **replicaMatrix**: a matrix of replicates with corresponding values (as generated by `generateReplicateMat`)
- **thresh**: the threshold for the p-values

P-values and the intensity values for each siRNA are stored in a text output file.

See Also

`hitselectionZscore`, `hitselectionZscorePval`, `Ttest`

Examples

```r
data(scoredDataset1, package="RNAither")
data(pValVec1, package="RNAither")

##for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

scoredHits1 <- hitselectionPval(scoredDataset1, pValVec1, "SigIntensity", "Pval_hits", 0.05, "GeneName", "pvalue_testfile1.txt")

newdataset <- scoredHits1[[1]]
hitvector <- scoredHits1[[2]]
```
hitselectionZscore  

Selecting hits according to ZScores

Description

Selects significant genes according to their ZScore.

Usage

hitselectionZscore(dataset, col4zscore, col4sel, thresh, flag, flag2, col4anno, sumFunc, file4hits)

Arguments

dataset  
an R data frame generated with `generateDatasetFile`

col4zscore  
a character vector specifying the name of the column containing the ZScores, usually SigIntensity

col4sel  
a character vector specifying the name of the new dataset column where hits will be stored

thresh  
the threshold for the ZScores. The interpretation depends on the choice of the parameter `flag2`.

flag  
1 or 2. 1 means the ZScores are kept per well, 2 that they are summarized according to the parameter `sumFunc`.

flag2  
1, 2 or -2.

- If 1 is chosen and `thresh == n`, then the n greatest Zscores are chosen as hits.
- If 1 is chosen and `thresh == -n`, then the n smallest Zscores are chosen.
- If 1 is chosen and `thresh == 0`, all ZScores are chosen and written to the output file.
- If 2 is chosen, all Zscores greater than or equal to `thresh` are chosen.
- If -2 is chosen, all Zscores smaller than or equal to `thresh` are chosen.

col4anno  
a character string specifying the name of the dataset column to be used to define the replicate, for example "GeneName" or "Internal_GeneID"

sumFunc  
the function used to summarize ZScore values, e.g. `mean` or `median`.

file4hits  
the name of the file to store the results in

Details

If `flag2 == -2`, and there are no ZScores under the defined threshold `thresh`, the threshold is increased to `min(ZScores)`.

If `flag2 == 2`, and there are no ZScores over the defined threshold `thresh`, the threshold is increased to `max(ZScores)`.

Value

A list containing:

dataset  
the dataset with an added column defining the hits in the form of a binary vector

hitVector  
the binary vector itself

thresh  
the threshold for the ZScores

ZScores are stored in a text output file.
References


See Also

hitselectionPval, hitselectionZscorePval, Ttest

Examples

data(scoredDataset1, package="RNAither")
data(pValVec1, package="RNAither")

## for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

scoredHits1 <- hitselectionZscore(scoredDataset1, "SigIntensity", "Zscore_hits", -10, 2, 1, "GeneName", median, "Zscores_testfile1.txt")

newdataset <- scoredHits1[[1]]
hitvector <- scoredHits1[[2]]

hitselectionZscorePval

Selecting hits according to ZScores and p-values

Description

Selects significant genes according to their ZScore (summarized with the gene median) and p-values.

Usage

hitselectionZscorePval(dataset, pValVec, col4zscore, col4sel, thresh, thresh2, flag2, col4anno, sumFunc, file4hits)

Arguments

dataset an R data frame generated with generateDatasetFile
pValVec a vector of p-values, as generated by one of the test functions Ttest, MannWhitney or RankProduct
col4zscore a character vector specifying the name of the column containing the ZScores, usually "SigIntensity"
col4sel a character vector specifying the name of the new dataset column where hits will be stored
thresh the threshold for the ZScores. The interpretation depends on the choice of the parameter flag2.
thresh2 the threshold for the p-values
flag2 2 or -2. If 2 is chosen, all Zscores greater than or equal to thresh are chosen. If -2 is chosen, all Zscores smaller than or equal to thresh are chosen.
hitselectionZscorePval

colanno | a character string specifying the name of the dataset column to be used to define the replicate, for example "GeneName" or "Internal_GeneID".

sumFunc | the function used to summarize ZScore values, e.g. mean or median.

file4hits | the name of the file to store the results in

Details

If there are no p-values under the defined threshold thresh2, it is increased to min(pvalvec).

If flag2 == -2 and there are no ZScores under the defined threshold thresh, it is increased to min(ZScores).

If flag2 == 2 and there are no ZScores over the defined threshold thresh, it is increased to max(ZScores).

If there are not hits for the combined threshold of p-values and ZScores, the ZScore threshold is changed until there is a hit.

Value

A list containing:

dataset | the dataset with an added column defining the hits in the form of a binary vector

hitVector | the binary vector itself

thresh | the threshold for the ZScores

thresh2 | the threshold for the p-values

ZScores and p-values are stored in a text output file.

See Also

hitselectionPval, hitselectionZscore, Ttest

Examples

data(scoredDataset1, package="RNAither")
data(pValVec1, package="RNAither")

## for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

scoredHits1 <- hitselectionZscorePval(scoredDataset1, pValVec1, "SigIntensity", "Zscore_pval_hits", -1.5, 0.05, -2, "GeneName", median, "Zscores_pvals_testfile1.txt")

newdataset <- scoredHits1[[1]]

hitvector <- scoredHits1[[2]]
Incorporate a vector of p-values into a dataset

Description
Incorporates a vector of p-values into a dataset. Also works with a dataset containing values per well (non summarized), or with a hit vector.

Usage
incorporatepValVec(dataset, pValVec, replicaMatrix, col4anno, colname4pval)

Arguments
- dataset: an R data frame generated with `generateDatasetFile`
- pValVec: a vector of p-values
- replicaMatrix: a matrix of replicate values, as generated by `generateReplicateMat`
- col4anno: a character string specifying the name of the dataset column to be used to define the replicate, for example "GeneName" or "Internal_GeneID"
- colname4pval: a character string specifying the name of the dataset column the p-values will be stored in

Value
Returns the dataset with an added column of p-values.

See Also
`multTestAdjust`, `Ttest`

Examples
```r
data(exampleDataset, package="RNAither")
data(scoredDataset1, package="RNAither")
##scoredDataset1 already contains the p-value column
data(pValVec1, package="RNAither")
##for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

temp <- generateReplicateMat(dataset, 1, "Intensities", "SigIntensity", "GeneName")
replicamatrix <- temp[[1]]
newdataset <- incorporatepValVec(dataset, pValVec1, replicamatrix, "GeneName", "pvals")
##newdataset and scoredDataset1 are now equivalent
```
indexSubset

**Description**

Used together with `createSubset`, returns the indexes in the main dataset of the wells chosen as a subset by the previous call of `createSubset`.

**Usage**

`indexSubset(listIDs, equalTo)`

**Arguments**

- `listIDs`: a character string and one of the following: Spotnumber, Internal_GeneID, GeneName, SpotType, SigIntensity, SDSIntensity, Background, LabtekNb, RowNb, ColNb, ScreenNb, NbCells, PercCells, ...
- `equalTo`: A value or character string specifying the value in the chosen column, e.g. all wells on plate 2

**Value**

An integer vector containing the indexes in the main dataset of the wells chosen as a subset by the previous call of `createSubset`.

**See Also**

`createSubset`

**Examples**

```r
data(exampleDataset, package="RNAither")
subset <- createSubset(dataset, dataset$LabtekNb, 2)
indexOfSubsetInDataset <- indexSubset(dataset$LabtekNb, 2)
```

joinDatasetFiles

**Description**

Merges two or more dataset files into one, with one common header.

**Usage**

`joinDatasetFiles(listOfFiles, nbOfLinesInHeader, newHead, outputFile)`

**Arguments**

- `listOfFiles`: a list of the names of the files to join
- `nbOfLinesInHeader`: typically 3
- `newHead`: the new header
- `outputFile`: the name of the file to save the header and concatenated dataset in
joinDatasets

See Also

generateDatasetFile, joinDatasetFiles

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
saveDataset(header, dataset, "save_testfile1.txt")

header[[1]] <- "external_experiment_name,Test screen"
header[[2]] <- "comments,contains twice Screen Nb 1"

joinDatasetFiles(list( "save_testfile1.txt", "save_testfile1.txt"), 3, header, "concatenated_testfile.txt")
**LiWongRank**  
*Li Wong rank / invariant probeset normalization*

**Description**

Performs a Li Wong rank / invariant probeset normalization (see References).

**Usage**

LiWongRank(header, dataset, listOfArgs)

**Arguments**

header the header of a dataset file generated with generateDatasetFile

dataset an R data frame generated with generateDatasetFile

listOfArgs a list containing:
- a character string specifying the column whose values will be used for normalization
- a character string specifying the name of the dataset column to be used for the computation of the siRNA/gene ranks

**Details**

For each plate type/layout in each experiment, generates a ranked list of siRNAs according to their intensity values. Only siRNAs occurring only once on the plate are allowed in the list. The normalization is performed only if all plate types have a maximum of 20

For each "unique" siRNA on a plate type, the variance of its ranks across plates is computed. A histogram of variances is plotted and allows the user to choose a threshold. A list of siRNAs with rank variances under the given threshold is then returned for each plate type so that the user can choose an siRNA to normalize the plate with.

**Value**

Returns a list containing:

header the new header (with an added entry about the normalization procedure in the comments)

dataset the new dataset with normalized values. The old values are saved in an extra column with the suffix ".old"

**References**


Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- LiWongRank(header, dataset, list("SigIntensity", "GeneName"))
newheader <- normres[[1]]
newdataset <- normres[[2]]

---

lowessNorm  

Lowess normalization

Description

Performs a plate-wise lowess normalization of the data.

Usage

lowessNorm(header, dataset, listOfArgs)

Arguments

header  

dataset  

listOfArgs  

a list containing:

- a character string specifying the column used as channel 1 (colname4ch1)
- a character string specifying the column used as channel 2 (colname4ch2)
- optionally: the smoother span (smSpan) of the lowess function. This gives the proportion of points which influence the smooth at each value. Larger values give more smoothness. Defaults to 2/3.

Value

Corrects intensity values in case the values of ch2 decrease with the increase of ch1 values.

Returns a list containing:

header  

dataset  

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- lowessNorm(header, dataset, list("NbCells","SigIntensity"))
newheader <- normres[[1]]
newdataset <- normres[[2]]
mainAnalysis

Wrapper function for full automated analysis

Description

Performs a standard analysis of the data (quality and statistics) from a dataset file.

Usage

mainAnalysis(header, dataset, flagForSameExp, listOfNormalizations, listOfArgs4norm, listOfStatTests, listOfArgs4stat, multTestAdj, hitScoringVec1, hitScoringVec2, posNegFlag, flag4Gsea, vecOfChannels, whichOnto)

Arguments

header                  the header of a dataset file generated with generateDatasetFile
dataset                 an R data frame generated with generateDatasetFile
flagForSameExp          either 0 or 1. If 1, all experiments defined in the column ScreenNb in the dataset file must have the same design (same type and same number of replicates - exact plate layout is irrelevant) so that Spearman’s correlation coefficient can be computed between experiments (each with summarized replicates)
listOfNormalizations    a list of the normalization function to apply. Can be LiWongRank, varAdjust, divNorm, quantileNormalization, BScore, ZScore, ZScorePerScreen, subtractBackground, lowessNorm, controlNorm
listOfArgs4norm         a list containing, for each element of listOfNormalizations, the arguments to be passed on
listOfStatTests         a list of the statistical tests to perform. Can be Ttest, MannWhitney, RankProduct
listOfArgs4stat         a list containing, for each element of listOfStatTests, the arguments to be passed on
multTestAdj             indicates the p-value correction for multiple testing - one of “holm”, “hochberg”, “hommel”, “bonferroni”, “BH”, “BY”, “fdr”, or “none” (Type ?p.adjust for details))
hitScoringVec1          a vector of length 3 indicating (in that order):
                         - scoring according to p-value (0: no, 1: yes)
                         - scoring according to ZScore with ZScore < threshold (0: no, 1: yes), or according to ZScore < threshold and p-value < hitScoringVec2[1] (2)
                         - scoring according to ZScore with ZScore > threshold (0: no, 1: yes), or according to ZScore > threshold and p-value < hitScoringVec2[1] (2).
                         If hitScoringVec1[2] or hitScoringVec1[3] are equal to 2, hitScoringVec1[1] must be equal to one, otherwise p-values will not be computed.
hitScoringVec2          a vector of length 3 indicating the thresholds for hitScoringVec1
posNegFlag              either 0 (no controls available) or 1 (controls available)
makeBoxplot4PlateType

flag4Gsea Can be:
- either 0: No GSEA analysis is performed
- or 1: A GSEA analysis is performed for each hit scoring method
- or a binary vector that allows to choose which hit scoring method(s) will be used for a GSEA analysis. Hit scoring methods are sorted as follows: first, hits are scored according to the p-values of each test specified in listOfStatTests. Then, if the option of scoring hits according to p-values and Intensities is chosen (see hitScoringVec1, for each test, a hit vector is generated. Finally, if the option of scoring hits according to Intensities only is chosen, hit vectors are generated for this option.

vecOfChannels a character vector containing the names of the channels to be used for quality plots, for example "SigIntensity" or "NbCells"

whichOnto one of the three GO hierarchies: "biological_process", "molecular_function" or "cellular_component" - used for the GSEA analysis

Value

Generates the html output files index.html and indexnorm.html containing the quality analysis of raw and normalized data, respectively, and stats.html, containing the statistical analysis. If several normalization methods are applied, an indexnorm file is generated after each.

Note

This function is deprecated and kept only for backwards compatibility. Please use the "rnaiter" function instead.

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

mainAnalysis(header, dataset, 0, list(controlNorm), list(list(1, 0, "SigIntensity", 1)), list(Ttest, MannWhitney), list(list("1", 1, "SigIntensity", "GeneName"), list("1", 1, "SigIntensity", "GeneName"), "none", c(1, 0, 0), c(0.05, 0, 0), 1, 0, c("SigIntensity", "NbCells"), "biological_process")

makeBoxplot4PlateType Generate a boxplot of the data per plate

Description

Generates a boxplot comparing the same plates in different experiments.

Usage

makeBoxplot4PlateType(header, dataset, channel, plotTitle, showPlot)
**makeBoxplotControls**

**Arguments**

- **header**: the header of a dataset file generated with `generateDatasetFile`
- **dataset**: an R data frame generated with `generateDatasetFile`
- **channel**: a character string specifying the column whose values will be used for the boxplot
- **plotTitle**: the plot title
- **showPlot**: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

For each plate type, a boxplot of intensity values per experiment will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle` and the number of the plate.

The function returns a list containing:

- **plotName**: the plotname
- **minOfPlates**: the number of the first experiment
- **numOfPlates**: the number of the last experiment

**See Also**

- `makeBoxplotControls`
- `makeBoxplotControlsPerScreen`
- `makeBoxplotControlsPerPlate`
- `makeBoxplotPerPlate`
- `makeBoxplotPerScreen`

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
makeBoxplot4PlateType(header, dataset, "SigIntensity", "Data vs. Controls", 1)
```

makeBoxplotControls **Generate a boxplot of the data vs. the controls**

**Description**

Generates a boxplot of intensity values of negative controls, positive controls and experimental data.

**Usage**

```r
makeBoxplotControls(header, dataset, channel, plotTitle, showPlot)
```

**Arguments**

- **header**: the header of a dataset file generated with `generateDatasetFile`
- **dataset**: an R data frame generated with `generateDatasetFile`
- **channel**: a character string specifying the column whose values will be used for the boxplot
- **plotTitle**: the plot title
- **showPlot**: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.
Value

A boxplot of intensity values of negative controls, positive controls and experimental data will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

The function returns the plot name.

See Also

makeBoxplotControlsPerScreen, makeBoxplotControlsPerPlate, makeBoxplotPerPlate, makeBoxplotPerScreen

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

makeBoxplotControls(header, dataset, "SigIntensity", "Data vs. Controls", 1)

makeBoxplotControlsPerPlate(header, dataset, channel, plotTitle, plotDesign, showPlot)

Description

Generates a boxplot of intensity values of negative controls, positive controls and experimental data for each plate of each experiment available in the dataset.

Usage

makeBoxplotControlsPerPlate(header, dataset, channel, plotTitle, plotDesign, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the column whose values will be used for the boxplot
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

For each experiment, a series of boxplots of intensity values of negative controls, positive controls and experimental data will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle and the number of the experiment.

The function returns a list containing:

plotName the plotname
makeBoxplotControlsPerScreen

\[
c(\text{minOfScreens}, \text{numOfScreens})
\]

a vector with the number of the first experiment and of the last experiment

\[
c(\text{minOfPlates}, \text{numOfPlates})
\]

a vector with the number of the first plate and the number of the last plate

See Also

makeBoxplotControls, makeBoxplotControlsPerScreen, makeBoxplotPerPlate, makeBoxplotPerScreen

Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
makeBoxplotControlsPerPlate(header, dataset, "SigIntensity", "Data vs. Controls", 1, 1)
makeBoxplotControlsPerScreen
```

**Description**

Generates a boxplot of intensity values of negative controls, positive controls and experimental data for each experiment available in the dataset.

**Usage**

```r
makeBoxplotControlsPerScreen(header, dataset, channel, plotTitle, plotDesign, showPlot)
```

**Arguments**

- `header` : the header of a dataset file generated with `generateDatasetFile`
- `dataset` : an R data frame generated with `generateDatasetFile`
- `channel` : a character string specifying the column whose values will be used for the boxplot
- `plotTitle` : the plot title
- `plotDesign` : 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
- `showPlot` : 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

A series of boxplots of intensity values of negative controls, positive controls and experimental data will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

The function returns a list containing:

- `plotName` : the plotname
- `minOfScreens` : the number of the first experiment
- `numOfScreens` : the number of the last experiment
makeBoxplotPerPlate

See Also

makeBoxplotControls, makeBoxplotControlsPerPlate, makeBoxplotPerPlate, makeBoxplotPerScreen

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

makeBoxplotControlsPerScreen(header, dataset, "SigIntensity", "Data vs. Controls", 1, 1)

makeBoxplotPerPlate Generate a boxplot of the data per plate

Description

Generates a boxplot of intensity values per plate for each experiment available in the dataset.

Usage

makeBoxplotPerPlate(header, dataset, channel, plotTitle, plotDesign, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the column whose values will be used for the boxplot
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

For each experiment, a boxplot of intensity values per plate will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.
The function returns a list containing:

plotName the plotname
minOfScreens the number of the first experiment
numOfScreens the number of the last experiment

See Also

makeBoxplotControls, makeBoxplotControlsPerPlate, makeBoxplotControlsPerScreen, makeBoxplotPerScreen
makeBoxplotPerScreen

Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
makeBoxplotPerPlate(header, dataset, "SigIntensity", "Data vs. Controls", 1, 1)
```

---

**makeBoxplotPerScreen**  
*Generate a boxplot of the data per experiment*

**Description**

Generates a boxplot of intensity values per experiment.

**Usage**

```
makeBoxplotPerScreen(header, dataset, channel, plotTitle, showPlot)
```

**Arguments**

- `header`  
  the header of a dataset file generated with `generateDatasetFile`
- `dataset`  
  an R data frame generated with `generateDatasetFile`
- `channel`  
  a character string specifying the column whose values will be used for the boxplot
- `plotTitle`  
  the plot title
- `showPlot`  
  0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

A boxplot of intensity values per experiment will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

**See Also**

`makeBoxplotControls`, `makeBoxplotControlsPerPlate`, `makeBoxplotControlsPerScreen`, `makeBoxplotPerPlate`

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
makeBoxplotPerScreen(header, dataset, "SigIntensity", "Data vs. Controls", 1)
```
MannWhitney

Perform a Mann-Whitney test

Description
Performs the non-parametric Mann-Whitney test on the intensity data.

Usage
MannWhitney(dataset, listofargs)

Arguments
- dataset: an R data frame generated with `generateDatasetFile`
- listofargs: a list containing:
  - "g" (greater) for significant increase, "l" (lower) for significant decrease, or "two.sided" for both
  - either a number indicating the true value of the mean, or a character string indicating the name of the gene to compare with
  - a character string specifying the column whose values will be used for the test
  - a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"

Value
Returns a list containing:
- pValVec: a named vector of p-values
- dataset: the dataset with an added column "p.value.mannwhitney"
- paste("pValue.mannwhitney", testType, sep="_")
  - the character string "p.value.mannwhitney" concatenated with the testType (first element of listofargs)
- "Mann-Whitney test"
  - the character string "Mann-Whitney test"

See Also
- Ttest, RankProduct

Examples
data.exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
pvals1 <- MannWhitney(dataset, list("l", median(dataset$SigIntensity, na.rm=TRUE), "SigIntensity", "GeneName"))
pValVec1 <- pvals1[[1]]
scoredDataset1 <- pvals1[[2]]
**multTestAdjust**

*Adjust p-values for multiple testing*

**Description**

Adjusts p-values for multiple testing.

**Usage**

```r
multTestAdjust(pValVec, adjustMethod)
```

**Arguments**

- `pValVec`: a vector of p-values
- `adjustMethod`: one of the following: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". For details type `?p.adjust`.

**Value**

Returns a vector of corrected p-values. Can be integrated into a dataframe with the function `incorporatepValVec`.

**See Also**

`incorporatepValVec`, `Ttest`

**Examples**

```r
data(pValVec1, package="RNAither")

# for details on the generation of pValVec1, see the example of the Ttest function linked above.
newpvalvec <- multTestAdjust(pValVec1,"fdr")
```

---

**numCellQualControl**

*Quality control of the number of cells*

**Description**

Plots a histogram of the cell number per well and allows the user to set an upper and a lower threshold so as to exclude wells from the analysis.

**Usage**

```r
numCellQualControl(DataSetFile, nbLinesHeader, plotTitle)
```

**Arguments**

- `DataSetFile`: a dataset file generated with `generateDatasetFile`
- `nbLinesHeader`: typically 3
- `plotTitle`: the plot title
printGeneIDs

Description

Prints out the list of wells under and over the predefined thresholds in the shell.

Saves a list of discarded siRNA values (if applicable) in a text file named after the experiment name specified in the header concatenated with either "numCellQualControl\_discarded\_higher.txt" or "numCellQualControl\_discarded\_lower.txt".

Saves the histogram with the applied thresholds in a pdf file named after the experiment name specified in the header concatenated with the plotTitle.

Overwrites the given DatasetFile with the new dataset.

See Also

percCellQualControl

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
saveDataset(header, dataset, "save_testfile1.txt")

numCellQualControl("save_testfile1.txt", 3, "Histogram of the number of cells")

orderGeneIDs

Order a dataset

Description

Orders dataset according to one of its columns.

Usage

orderGeneIDs(dataset, ID1)

Arguments

dataset an R data frame generated with generateDatasetFile
ID1 a character string specifying the name of the column according to which the dataset will be sorted

Value

An R data frame (‘dataset’) ordered according to its values in the specified column.

See Also

order

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

orderedDataset=orderGeneIDs(dataset,"SigIntensity")
**percCellQualControl**  
*Quality control of the percentage of cells*

**Description**

Plots a histogram of the percentage of cells per well (ratio of the number of identified cells and the number of identified objects) and allows the user to set an upper and a lower threshold so as to exclude wells from the analysis.

**Usage**

`percCellQualControl(DataSetFile, nbLinesHeader, plotTitle)`

**Arguments**

- `DataSetFile`: a dataset file generated with `generateDatasetFile`
- `nbLinesHeader`: typically 3
- `plotTitle`: the plot title

**Value**

Prints out the list of wells under and over the predefined thresholds in the shell.

Saves a list of discarded siRNA values (if applicable) in a text file named after the experiment name specified in the header concatenated with either "percCellQualControl\_discarded\_higher.txt" or "percCellQualControl\_discarded\_lower.txt".

Saves the histogram with the applied thresholds in a pdf file named after the experiment name specified in the header concatenated with the `plotTitle`.

Overwrites the given `DataSetFile` with the new dataset.

**See Also**

`numCellQualControl`

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
saveDataset(header, dataset, "save_testfile1.txt")
percCellQualControl("save_testfile1.txt", 3, "Histogram of the number of cells")
```
plotBar

Plot signal intensities per well

Description

Plots signal intensity values for each well, a blue line showing the median, two green lines showing one median absolute deviation, two red lines showing two median absolute deviations.

Usage

plotBar(header, dataset, col4val, flag, plotTitle, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
col4val a character string specifying the column whose intensity values will be used for the plot
flag 0, 1, or 2. 0 uses the data from the complete dataset, 1 generates one plot for each experiment, 2 generates one plot for each plate.
plotTitle the plot title
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Saves the plots in pdf and png files named after the experiment name specified in the header concatenated with the plotTitle and if applicable the experiment number and/or the plate number.
When flag == 0, returns the plot name (plotName).
When flag == 1, returns a list containing:
plotName The plot name
minOfScreens The number of the first experiment
numOfScreens The number of the last experiment
When flag == 2, returns a list containing: the plot name, a vector with the number of the first experiment and of the last experiment, and a vector with the number of the first plate and the number of the last plate.

See Also

ZScorePlot, ZScorePlotTwo

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotname <- plotBar(header, dataset, "SigIntensity", 0, "Data per well", 1)
**plotControlHisto**  
*Plot a histogram of the data values and controls*

**Description**

Plots and saves a histogram of data values and shows the controls, if available, in color.

**Usage**

```r
plotControlHisto(header, dataset, channel, plotTitle, showPlot)
```

**Arguments**

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `channel`: a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
- `plotTitle`: the plot title
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

Saves the histogram in a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

Positive controls are plotted in green, negative controls in red.

The function returns the plot name.

**See Also**

`plotControlHistoPerplate`, `plotControlHistoPerscreen`

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
plotControlHisto(header, dataset, "SigIntensity", "Distribution of Data and Controls", 1)
```
plotControlHistoPerplate

Plot a histogram of the data values and controls per plate

Description

Plots and saves a histogram of data values per experiment and per plate and shows the controls, if available, in color.

Usage

plotControlHistoPerplate(header, dataset, channel, plotTitle, plotDesign, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Saves the histograms in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.
Positive controls are plotted in green, negative controls in red.
The function returns a list containing:

histoName the plotname
c(minOfScreens, numOfScreens) a vector with the number of the first experiment and of the last experiment
c(minOfPlates, numOfPlates) a vector with the number of the first plate and the number of the last plate

See Also

plotControlHisto, plotControlHistoPerscreen

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotControlHistoPerplate(header, dataset, "SigIntensity", "Distribution of Data and Controls", 1, 1)
plotControlHistoPerscreen

Plot a histogram of the data values and controls per experiment

Description

Plots and saves a histogram of data values per experiment and shows the controls, if available, in color.

Usage

plotControlHistoPerscreen(header, dataset, channel, plotTitle, plotDesign, showPlot)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Saves the histograms in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

Positive controls are plotted in green, negative controls in red.

The function returns a list containing:

histoName the plotname
minOfScreens the number of the first experiment
numOfScreens the number of the last experiment

See Also

plotControlHisto, plotControlHistoPerplate

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotControlHistoPerscreen(header, dataset, "SigIntensity", "Distribution of Data and Controls", 1, 1)
plotHisto

Plot a histogram of the data values

Description
Plots and saves a histogram of the chosen data values.

Usage
plotHisto(header, dataset, channel, plotTitle, showPlot)

Arguments
- header: the header of a dataset file generated with generateDatasetFile
- dataset: an R data frame generated with generateDatasetFile
- channel: a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
- plotTitle: the plot title
- showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value
Saves the histogram in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.
The function returns the plot name.

See Also
plotHistoPerplate, plotHistoPerscreen

Examples
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
plotHisto(header, dataset, "SigIntensity", "Distribution of Data and Controls", 1)

plotHistoPerplate

Plot a histogram of the data values per plate

Description
Plots and saves a histogram of the chosen data values per experiment and per plate.

Usage
plotHistoPerplate(header, dataset, channel, plotTitle, plotDesign, showPlot)
plotHistoPerscreen

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
channel a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
plotTitle the plot title
plotDesign 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Saves the histograms in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

The function returns a list containing:

histoName the plotname
c(minOfScreens, numOfScreens) a vector with the number of the first experiment and of the last experiment
c(minOfPlates, numOfPlates) a vector with the number of the first plate and the number of the last plate

See Also

plotHisto, plotHistoPerscreen

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotHistoPerplate(header, dataset, "SigIntensity", "Distribution of Data and Controls", 1, 1)

plotHistoPerscreen Plot a histogram of the data values per experiment

Description

Plots and saves a histogram of the chosen data values.

Usage

plotHistoPerscreen(header, dataset, channel, plotTitle, plotDesign, showPlot)
plotQQ

Make a QQ plot

Description

Shows and saves a QQ plot of the data.

Usage

plotQQ(header, dataset, channel, plotTitle, showPlot)
**plotQQperplate**

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>header</td>
<td>the header of a dataset file generated with <code>generateDatasetFile</code></td>
</tr>
<tr>
<td>dataset</td>
<td>an R data frame generated with <code>generateDatasetFile</code></td>
</tr>
<tr>
<td>channel</td>
<td>a character string specifying the name of the column containing the values to be plotted, e.g. &quot;SigIntensity&quot;</td>
</tr>
<tr>
<td>plotTitle</td>
<td>the plot title</td>
</tr>
<tr>
<td>showPlot</td>
<td>0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.</td>
</tr>
</tbody>
</table>

### Value

Saves the QQ plot in a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

The function returns the plot name.

### See Also

`plotQQperscreen, plotQQperplate`

### Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotQQ(header, dataset, "SigIntensity", "QQplot", 1)
```

### Make a QQ plot per plate

**Description**

Shows and saves a QQ plot of the data for each experiment and each plate in the dataset.

**Usage**

`plotQQperplate(header, dataset, channel, plotTitle, plotDesign, showPlot)`

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>header</td>
<td>the header of a dataset file generated with <code>generateDatasetFile</code></td>
</tr>
<tr>
<td>dataset</td>
<td>an R data frame generated with <code>generateDatasetFile</code></td>
</tr>
<tr>
<td>channel</td>
<td>a character string specifying the name of the column containing the values to be plotted, e.g. &quot;SigIntensity&quot;</td>
</tr>
<tr>
<td>plotTitle</td>
<td>the plot title</td>
</tr>
<tr>
<td>plotDesign</td>
<td>1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.</td>
</tr>
<tr>
<td>showPlot</td>
<td>0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.</td>
</tr>
</tbody>
</table>
plotQQperscreen

Value

Saves the QQ plots in a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

The function returns a list containing:

- `histoName`: the plotname
- `c(minOfScreens, numOf Screens)`: a vector with the number of the first experiment and of the last experiment
- `c(minOfPlates, numOfPlates)`: a vector with the number of the first plate and the number of the last plate

See Also

`plotQQ, plotQQperscreen`

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotQQperplate(header, dataset, "SigIntensity", "QQplot", 1, 1)

plotQQperscreen

Make a QQ plot per experiment

Description

Shows and saves a QQ plot of the data for each experiment in the dataset.

Usage

`plotQQperscreen(header, dataset, channel, plotTitle, plotDesign, showPlot)`

Arguments

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `channel`: a character string specifying the name of the column containing the values to be plotted, e.g. "SigIntensity"
- `plotTitle`: the plot title
- `plotDesign`: 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.
pValVec1

Value

Saves the QQ plots in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

The function returns a list containing:

histoName      the plotname
minOfScreens   the number of the first experiment
numOfScreens   the number of the last experiment

See Also

plotQQ, plotQQperplate

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotQQperscreen(header, dataset, "SigIntensity", "QQplot", 1, 1)

pValVec1  A vector of p-values after a median normalization and a t-test

Description

See divNorm and Ttest for details

Usage

pValVec1

Format

vector

pValVec2  A vector of p-values after a Mann-Whitney test

Description

See MannWhitney for details

Usage

pValVec2

Format

vector
quantileNormalization  

Quantile normalization

Description
Quantile normalization (see References)

Usage
quantileNormalization(header, dataset, listOfArgs)

Arguments
header  
the header of a dataset file generated with generateDatasetFile

dataset  
an R data frame generated with generateDatasetFile

listOfArgs  
a list containing:
- a character string specifying the column whose values will be used for normalization
- 1 or 2, 1 meaning a normalization per experiment, 2 meaning a normalization per plate

Value
Returns a list, containing:

header  
the new header (with an added entry about the normalization procedure in the comments)

dataset  
the new dataset with normalized values. The old values are saved in an extra column with the suffix ".old"

References

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normres <- quantileNormalization(header, dataset, list(2, "SigIntensity"))
newheader <- normres[[1]]
newdataset <- normres[[2]]
RankProduct

Perform a Rank Product test

Description

Performs the non-parametric rank product test on the intensity data.

Usage

RankProduct(dataset, listofargs)

Arguments

dataset an R data frame generated with generateDatasetFile
listofargs a list containing:
- the number of permutations to perform to compute the p-values (usually 100)
- 1 or 2, depending if the search is for a significant decrease or increase
- a character string specifying the column whose values will be used for the test
- a character string specifying the name of the dataset column to be used to define the replicate, for example "GeneName" or "Internal_GeneID"

Value

Returns a list containing

pValVec a named vector of p-values
dataset the dataset with an added column "p.value.rankproduct"
paste("pValue.rankproduct", testType, sep="_")
the character string "p.value.rankproduct"
"Rank product test"
the character string "Rank product test"

The p values returned are equivalent to the percentage of false prediction (pfp), which in theory is the equivalent of false discovery rate (FDR). It is possible that they are larger than 1.

See Also

Ttest, MannWhitney

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

pvals1 <- RankProduct(dataset, list(100, 1, "SigIntensity", "GeneName"))
pValVec1 <- pvals1[[1]]
scoredDataset1 <- pvals1[[2]]
replicatesCV  

*Compute the correlation of variation (CV)*

**Description**

Computes the correlation of variation as defined in Tseng et al. (see References)

**Usage**

```r
replicatesCV(header, dataset, PlotTitle, col4val, col4anno, plotDesign, showPlot)
```

**Arguments**

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `PlotTitle`: the plot title
- `col4val`: a character string specifying the column whose values will be used to compute the correlation of variation
- `col4anno`: a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
- `plotDesign`: 1 or 2. 1 will generate one window containing all plots, 2 will generate a series of plots.
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

The correlation of variation of an siRNA is defined as the standard deviation of its values divided by their mean.

The function generates a plot of the average intensity against the CV for each experiment. The plot will be saved as a pdf and a png file named after the experiment name specified in the header concatenated with the `PlotTitle`.

The function returns a list containing:

- `histoName`: the plotname
- `minOfScreens`: the number of the first experiment
- `numOfScreens`: the number of the last experiment

**References**


**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
replicatesCV(header, dataset, "Correlation of Variation versus Mean Intensity", "SigIntensity", "GeneName", 1, 0)
```
replicatesSpearmancor  

Compute the correlation coefficient between replicates or experiments

Description

Computes Spearman’s rank correlation coefficient for each replicate - either inside each experiment, or between experiments.

Usage

replicatesSpearmancor(header, dataset, flag, col4val, col4anno, fileNameSuffix)

Arguments

header the header of a dataset file generated with generateDatasetFile

dataset an R data frame generated with generateDatasetFile

flag 1 or 2. 1 will compute the coefficient for a maximum of 3 replicates, for each experiment available in the dataset. 2 will summarize the replicates from each experiment with their root mean square and compute the correlation coefficient between experiments.

col4val a character string specifying the column whose values will be used to compute the correlation coefficient

col4anno a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"

fileNameSuffix a character string that will be used to name the output file containing a table with the correlation coefficients.

Value

For flag==1, the correlation coefficients are printed out to the shell and saved in a text file named after the experiment name specified in the header concatenated with the character string filename-suffix and "Spearmancor.txt".

For flag==2, the correlation coefficients are printed out to the shell and saved in a text file named after the experiment name specified in the header concatenated with the character string filename-suffix and "Spearmancor\_AllExp.txt".

The function returns a table containing the correlation coefficients.

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

replicatesSpearmancor(header, dataset, 1, "SigIntensity", "GeneName", "testfile1")
**rms**  
*Compute the replicate root mean square*

**Description**
Computes the root mean square of replicate values

**Usage**
rms(Ivec, na.rm = T)

**Arguments**
- **Ivec**  
  All channel values for a specific siRNA/gene
- **na.rm**  
  Removes NA values

**Value**
A double giving the root mean square of the given replicate values.

**See Also**
- trim, closestToZero, furthestFromZero, summarizeReps, summarizeRepsNoFiltering

**Examples**
data(exampleDataset, package="RNAither")
Indexes <- findReplicates(dataset, "GeneName", "CPSF1")
rmsval <- rms(dataset$SigIntensity[Indexes])

---

**rnaither**  
*Wrapper function for full automated analysis*

**Description**
Performs a standard analysis of the data (quality and statistics) from a dataset file.

**Usage**
rnaither(data, expname, excludeCellcounts="none", logtransform=FALSE, normalization=c("lowess","bscore"), test="ttest", scorethresh=2.0, pvalthresh=0.05, dogo=FALSE, outdir="results", layoutnames="NA", makeplots=TRUE, reorder=TRUE)
Arguments

**data**
A data frame containing the experimental data to analyze. Each row is corresponding to one well, with the following columns:
- **Spotnumber** The position of the well on the plate
- **Internal_GeneID** The ID of the siRNA
- **GeneName** The gene name
- **SpotType** Can be -1, 0, 1 or 2. Type -1 wells (e.g. empty wells, wells with poor quality) are not considered in subsequent analyses but are kept in the data set for the sake of completeness. Type 0 wells correspond to negative controls, type 1 wells to positive controls. Type 2 wells correspond to the standard data wells.
- **SigIntensity** The signal intensity (channel 1)
- **SDSIntensity** The standard deviation of the signal intensity, if available
- **Background** The background per well, if available
- **LabtekNb** The plate number
- **RowNb** The row number
- **ColNb** The column number
- **ScreenNb** The screen number
- **NbCells** E.g. the number of cells identified in the well (channel 2)
- **PercCells** The ratio (number of identified cells)/(number of identified objects)

**expname**
A character string, assigning a name to the experiment. This will be used as title in the html output generated by rnaither.

**excludeCellcounts**
a string constant, one of "none", "lowest", "both", "lowestperplate" or "bothperplate". The default is "none". This parameter can be used to exclude wells from the analysis that have very low or very high numbers of cells.
- "none" No wells will be excluded based on the number of cells they contain.
- "lowest", "lowestperplate" The wells with the lowest 5 percent of cellcounts will be excluded from further analysis. "lowest" will consider the entire screen at once, and exclude the wells that are overall the lowest 5 percent. "lowestperplate" will consider each plate separately, excluding on each plate the 5 percent of wells having the lowest cellcounts.
- "both", "bothperplate" The wells with the lowest and highest 5 percent of cellcounts will be excluded from further analysis. Excluding wells with high cell counts may be useful for image based screens, if it is suspected that cells overlap in images, which might cause problems for image processing. "both" will consider the entire screen at once, and exclude the wells that are overall the lowest and highest 5 percent. "bothperplate" will consider each plate separately, excluding on each plate the 5 percent of wells having the lowest and highest cellcounts.

**logtransform**
A logical variable, specifying whether or not the signal intensities should be log-transformed. Default is FALSE.

**normalization**
A list of strings containing the normalization steps to carry out. The default are is c("lowess","bscore"). The following normalization procedures are available:
- "lowess" To carry out lowess normalization. This corrects for effects of cell counts on the signal intensities.
- "liwong" To carry out Li-Wong rank normalization of the signal intensities.
- "varadjust" To divide each signal intensity value by the variance of the signal intensities on the respective plate.
- "divnorm" To divide each signal intensity value by the median signal intensity of the respective plate.
- "quantile" To carry out a quantile normalization on the signal intensities.
- "bscore" To carry out a bscore normalization on the signal intensities (corrects for spatial effects on a plate).
- "zscore" To carry out a zscore normalization (subtract median of plate, divide by median absolute deviation per plate).
- "negcontrol" To normalize on the negative controls - subtract median of negative controls, divide by MAD of negative controls, per plate.
- "percontrol" To do a percentage of controls normalization - Rescale signal intensities so that mean of negative controls is 100, mean of positive controls is 0.
- "percneg" To do a percentage of negative controls normalization - set mean of negative controls to 100, zero signal intensity remains at 0. Normalization routines will be executed in the order as they occur in the list.

test
Specify what statistical test should be used to identify hits. One of
- "ttest" to carry out a t-test if the mean score for a given siRNA / Gene is 0.
- "wilcox" to carry out a Wilcoxon test if the mean score for a given siRNA / Gene is 0.
- "none" to carry out no statistical test.
The default is "ttest".

scorethresh
The threshold on the normalized score to be used to identify hits. The default is 2.0, hence siRNAs with score > +2 or score < -2 are considered hits.
pvalthresh
The threshold on the p-value from the statistical test to be used to identify hits. The default is 0.05
dogo
A logical variable, specifies whether or not a Gene Ontology-based analysis should be carried out. This parameter is currently ignored, GO is presently not supported by the rnaither wrapper.
outdir
A string specifying the directory in which the results should be stored. Can be an absolute or relative path.
layoutnames
A list of strings, that can be used to assign names to different layouts in the screen. The list should contain the same number of elements as there are different layouts in the screen. These names will be used as labels for the layouts in the html output. If this parameter is not specified, layouts will be numbered in the canonical way.
makeplots
TRUE or FALSE, if set to FALSE, only a subset of the quality control plots will be generated. This speeds up processing, but will result in missing images in the html output.
reorder
A logical variable, indicating whether dataset should be reordered prior to processing further. This is recommended if the data frame is incomplete, i.e. if wells or plates are missing completely. reorder=T will considerably slow down the analysis.

Value
Generates the html output file index.html in the directory specified by the outdir parameter.
**saveDataset**

Save the normalized dataset into a dataset text file

**Description**

Saves the normalized dataset and corresponding header into the specified dataset text file.

**Usage**

`saveDataset(header, data, dataSetFile)`

**Arguments**

- **header**
  
  the header of a dataset file generated with `generateDatasetFile`

- **data**
  
  an R data frame generated with `generateDatasetFile`

- **dataSetFile**
  
  the name of the text file the data will be saved in; can be the same as the old file (will be overwritten without prompting)

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normres <- divNorm(header, dataset, list(median, 2, 1, "SigIntensity", 1))
nnewheader <- normres[[1]]
nnewdataset <- normres[[2]]
saveDataset(nnewheader, nnewdataset, "save_testfile1.txt")
```

**saveOldIntensityColumns**

Save old intensity value columns

**Description**

Duplicates the specified column and adds it to the end of the dataset.

**Usage**

`saveOldIntensityColumns(dataset, col4val)`

**Arguments**

- **dataset**
  
  an R data frame generated with `generateDatasetFile`

- **col4val**
  
  a character string specifying the column whose values will be saved as an extra column before normalization
The values in the chosen column are saved in an extra column with the suffix ".old".

Examples

```r
data(exampleDataset, package="RNAither")
newdataset <- saveOldIntensityColumns(dataset, "SigIntensity")
```

### savepValVec

Save p-values to file

**Description**

Saves a vector of p-values to a text file.

**Usage**

```r
savepValVec(pValVec, filename)
```

**Arguments**

- `pValVec`: a vector of p-values
- `filename`: the name of the text file to save the p-values to.

**See Also**

*Ttest*

**Examples**

```r
data(pValVec1, package="RNAither")
```

### scoredDataset1

A dataset containing an additional column showing the p-values, after a median normalization and a t-test

**Description**

See `divNorm` and `Ttest` for details

**Usage**

```r
scoredDataset1
```

**Format**

see `generateDatasetFile` for details
scoredDataset2

A dataset containing an additional column showing the p-values after a Mann-Whitney test

Description

See MannWhitney for details

Usage

scoredDataset1

Format

see generateDatasetFile for details

SNRQualControl

Computing the SNR

Description

Computes the signal to noise ratio for all data, per experiment and per plate for a complete dataset file and plots histograms of the results.

Usage

SNRQualControl(dataSetFile, nbLinesHeader, channel, noise, plotTitle, showPlot)

Arguments

dataSetFile a dataset file generated with generateDatasetFile

nbLinesHeader typically 3

channel a character string specifying the name of the column containing the values for computing the SNR, e.g. "SigIntensity"

noise A character string specifying the name of the column containing the values for computing the SNR, e.g. "Background"

plotTitle the plot title

showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows

Value

Shows histogram plots of the SNR for the whole dataset file, per experiment and per plate and saves them in a pdf file. The name of the file will be the concatenation of the experiment name specified in the header and the function argument plotTitle.
Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
saveDataset(header, dataset, "save_testfile1.txt")

SNRQualControl("save_testfile1.txt", 3, "SigIntensity", "Background", "SNR", 1)
```

```
spatialDistrib

Generate spatial plots of intensity values
```

Description

Generate plots of plates and their intensity values.

Usage

```r
spatialDistrib(header, dataset, plotTitle, col4plot, col4anno, showPlot)
```

Arguments

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `plotTitle`: the plot title
- `col4plot`: a character string specifying the column whose values will be used for the plot
- `col4anno`: a character string specifying the column whose values will be used for the annotation of the plot
- `showPlot`: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

For each plate, the plot will be saved as a png file named after the experiment name specified in the header concatenated with the `plotTitle`, the number of the experiment, and the number of the plate.

Wells containing positive controls are marked with a "P", wells containing negative controls with an "N".

Each plate will also be saved as an html file containing mouse-overs with the siRNA name for each well.

The function returns a list containing:

- `histoName`: the plotname
- `c(minOfScreens, numOfScreens)`: a vector with the number of the first experiment and of the last experiment
- `c(minOfPlates, numOfPlates)`: a vector with the number of the first plate and the number of the last plate

See Also

`compareReplicateSD`, `compareReplicateSDPerScreen`
Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

spatialDistrib(header, dataset, "Spatial distribution of cell counts", "NbCells", "GeneName", 1)

spatialDistribHits

Description

Plots the plates showing the spatial distribution of the hits using the `plotPlate` function of the prada package.

Usage

spatialDistribHits(header, dataset, plotTitle, col4hits, col4anno, showPlot)

Arguments

- `header` the header of a dataset file generated with `generateDatasetFile`
- `dataset` an R data frame generated with `generateDatasetFile`
- `plotTitle` the plot title
- `col4hits` a character vector specifying the name of the dataset column containing the binary hit vector
- `col4anno` a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
- `showPlot` 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

For each plate, the plot will be saved as a png file named after the experiment name specified in the header concatenated with the `plotTitle`, the number of the experiment, and the number of the plate.

Wells containing positive controls are marked with a "P", wells containing negative controls with an "N".

Each plate will also be saved as an html file containing mouse-overs with the siRNA name for each well.

The function returns a list containing:

- `histoName` the plotname
- `c(minOfScreens, numOfScreens)` a vector with the number of the first experiment and of the last experiment
- `c(minOfPlates, numOfPlates)` a vector with the number of the first plate and the number of the last plate
substractBackground

Description

Substracts a specified background value from the intensity values.

Usage

substractBackground(header, dataset, listOfArgs)

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
listOfArgs a list containing:
- a character string specifying the column whose values will be used for back-
ground substraction
- a character string specifying the column whose values will be used as back-
ground

Value

A list containing:

header The new header (with an added entry about the normalization procedure in the comments)
dataset The new dataset with normalized values. The old values are saved in an extra
column of the dataset with the suffix ".old"
**sumChannels**

Summarizes two channels, for example by computing their ratio.

### Usage

```r
sumChannels(header, dataset, funName, colname4ch1, colname4ch2)
```

### Arguments

- `header`: the header of a dataset file generated with `generateDatasetFile`
- `dataset`: an R data frame generated with `generateDatasetFile`
- `funName`: the function used to summarize the two channels, for example `divideChannels`
- `colname4ch1`: a character string specifying the name of the dataset column containing the first channel
- `colname4ch2`: a character string specifying the name of the dataset column containing the second channel

### Details

The original dataset columns are saved as extra columns with the suffix ".old" by the function `saveOldIntensityColumns`.

### Value

A list containing:

- `header`: the header with an entry about the channel summarization added in the comments section
- `newDataset`: the new dataset

### See Also

`eraseDataSetColumn, divideChannels, saveOldIntensityColumns`

### Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- subtractBackground(header, dataset, list("SigIntensity", "Background"))
newheader <- normres[[1]]
newdataset <- normres[[2]]

newdataset = sumChannels(header, dataset, divideChannels, "SigIntensity", "NbCells")
```
summarizeReps

Generate a new dataset with summarized replicates

Description

Generates a new dataset with summarized replicates.

Usage

summarizeReps(data, funSum, col4val, col4anno, cols2del)

Arguments

data
  an R data frame generated with generateDatasetFile
funSum
  a function used to summarize the values of a replicate, e.g. mean, median, rms, trim, max, min, closestToZero, furthestFromZero, ...
col4val
  a character vector (containing for example "SigIntensity", Background, NbCells, PercCells,...) specifying the columns that will be summarized by funSum
col4anno
  a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
cols2del
  a character vector containing the columns to delete, for example "SDSIntensity"

Details

All columns containing replicate values will be summarized by funSum. For all columns containing positions, screen numbers, plate numbers, etc., all information for different replicates will be kept, comma-separated. All columns containing standard deviations of channels should be specified in colnames2delete.

Value

Returns the summarized dataset.

See Also

summarizeRepsNoFiltering, eraseDataSetColumn, generateReplicateMat, generateRepMatNoFilter, mean, median, rms, trim, max, min, closestToZero, furthestFromZero

Examples

data(exampleDataset, package="RNAither")

colname4val <- c("SigIntensity", "Background", "NbCells", "PercCells")
summarizeddataset <- summarizeReps(dataset, mean, colname4val, "GeneName", "SDSIntensity")
summarizeRepsNoFiltering

Generate a new dataset with summarized replicates

Description

Generates a new dataset with summarized replicates. Keeps wells/spots with SpotType -1 in the dataset, but intensity values are replaced with NA.

Usage

summarizeRepsNoFiltering(data, funSum, col4val, col4anno, cols2del)

Arguments

data an R data frame generated with generateDatasetFile
funSum a function used to summarize the values of a replicate, e.g. mean, median, rms, trim, max, min, closestToZero, furthestFromZero, ...
col4val a character vector (containing for example "SigIntensity", Background, NbCells, PercCells,...) specifying the columns that will be summarized by funSum
col4anno a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"
cols2del a character vector containing the columns to delete, for example "SDSIntensity"

Details

All columns containing replicate values will be summarized by funSum. For all columns containing positions, screen numbers, plate numbers, etc., all information for different replicates will be kept, comma-separated. All columns containing standard deviations of channels should be specified in colnames2delete.

Value

Returns the summarized dataset.

See Also

summarizeReps, eraseDataSetColumn, generateReplicateMat, generateRepMatNoFilter, mean, median, rms, trim, max, min, closestToZero, furthestFromZero

Examples

data(exampleDataset, package="RNAither")
colname4val <- c("SigIntensity", "Background", "NbCells", "PercCells")
summarizeddataset <- summarizeRepsNoFiltering(dataset, mean, colname4val, "GeneName", "SDSIntensity")
**trim**

*Compute the replicate mean with trimmed values*

**Description**

Computes the mean of replicate values, omitting the highest and the lowest 5.

**Usage**

```r
test(arriveT, na.rm = T)
```

**Arguments**

- **arriveT**: All channel values for a specific siRNA/gene.
- **na.rm**: Removes NA values.

**Value**

A double giving the trimmed mean of the given replicate values, i.e. omitting the highest and the lowest 5.

**See Also**

`rms`, `closestToZero`, `furthestFromZero`, `summarizeReps`, `summarizeRepsNoFiltering`

**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

Indexes <- findReplicates(dataset, "GeneName", "CPSF1")
replicatemean <- trim(dataset$SigIntensity[Indexes])
```

---

**Ttest**

*Perform a Student’s t-test*

**Description**

Performs a Student’s t-test on the intensity data.

**Usage**

```r
Ttest(dataset, listofargs)
```
varAdjust

Arguments

dataset an R data frame generated with `generateDatasetFile`
listofargs a list containing:
  - "g" (greater) for significant increase, "l" (lower) for significant decrease, or "two.sided" for both
  - either a number indicating the true value of the mean, or a character string indicating the name of the gene to compare with
  - a character string specifying the column whose values will be used for the test
  - a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID"

Value

Returns a list containing:

pValVec a named vector of p-values
dataset the dataset with an added column "p.value.mannwhitney"
paste("pValue.ttest", testType, sep="_") the character string "pValue.ttest" concatenated with the testType (first element of listofargs)
"t test" the character string "t test"

See Also

MannWhitney, RankProduct

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
pvals1 <- Ttest(dataset, list("l", median(dataset$SigIntensity, na.rm=TRUE), "SigIntensity", "GeneName"))
pValVec1 <- pvals1[[1]]
scoredDataset1 <- pvals1[[2]]
Arguments

header the header of a dataset file generated with `generateDatasetFile`
dataset an R data frame generated with `generateDatasetFile`
listOfArgs a list containing:
- a character string specifying the column whose values will be used for normalization
- 1 or 2, 1 meaning a normalization per screen, 2 a normalization per plate
- a flag specifying whether controls should be excluded for the computation of the median absolute deviation (1) or not (0).

Value

Divides the intensity values by their median absolute deviation (of the experiment or of the plate).
Returns a list containing:

header The new header (with an added entry about the normalization procedure in the comments)
dataset The new dataset with normalized values. The old values are saved in an extra column of the dataset with the suffix ".old"

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normres <- varAdjust(header, dataset, list(1, "SigIntensity", 1))
newheader <- normres[[1]]
newdataset <- normres[[2]]

vennDiag

Plotting a Venn Diagram to compare hits

Description

Plots a Venn Diagram of up to three binary hit vectors.

Usage

```
vennDiag(header, listOfCols, listOfNames, plotTitle, showPlot)
```

Arguments

header the header of a dataset file generated with `generateDatasetFile`
listOfCols a list of binary hit vectors to compare
listOfNames a list of character strings for the annotation of the Venn Diagram
plotTitle the plot title
showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows
Value

The plot is saved in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

The function returns the plot name.

See Also

Ttest, MannWhitney

Examples

data(exampleHeader, package="RNAither")
data(pValVec1, package="RNAither")
data(pValVec2, package="RNAither")
data(scoredDataset1, package="RNAither")
data(scoredDataset2, package="RNAither")

##for details on the generation of pValVec and scoredDataset, see the examples of the functions Ttest and MannWhitney linked above.

scoredHits1 <- hitselectionPval(scoredDataset1, pValVec1, "SigIntensity", "pValue.ttest_l", 0.05, "GeneName", "pvalue_testfile1.txt")
scoredHits2 <- hitselectionPval(scoredDataset2, pValVec2, "SigIntensity", "pValue.mannwhitney_l", 0.05, "GeneName", "pvalue_testfile2.txt")

hitvector1 <- scoredHits1[[2]]
hitvector2 <- scoredHits2[[2]]

plot_name <- vennDiag(header, list(hitvector1, hitvector2), list("t test", "Mann-Whitney test"), "Venn diagram", 1)

---

volcanoPlot  Making a volcano plot

Description

Makes a volcano plot of the data.

Usage

volcanoPlot(header, dataset, col4plotx, col4ploty, col4anno, plotTitle, sigLevel, showPlot)

Arguments

header  the header of a dataset file generated with generateDatasetFile

dataset  an R data frame generated with generateDatasetFile

col4plotx  a character vector specifying the name of the column containing the intensity values, usually SigIntensity

col4ploty  a character vector specifying the name of the dataset column containing the corresponding p-values
ZPRIMEQualControl

co14anno a character string specifying the name of the dataset column to be used to define the replicate, e.g. "GeneName" or "Internal_GeneID".

plotTitle the plot title

sigLevel the significance level for the p-value, indicating where a horizontal green line will be drawn

showPlot 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Plots the intensity values against the negative decadic logarithm of the p-values. A green horizontal line is drawn at the specified significance level.

The plot is saved in a pdf and a png file named after the experiment name specified in the header concatenated with the plotTitle.

The function returns the plot name.

See Also

Ttest

Examples

data(exampleHeader, package="RNAither")
data(pValVec1, package="RNAither")
data(scoredDataset1, package="RNAither")

##for details on the generation of pValVec1 and scoredDataset1, see the example of the Ttest function linked above.

scoredHits1 <- hitselectionPval(scoredDataset1, pValVec1, "SigIntensity", "pValue.ttest_1", 0.05, "GeneName", "pvalue_testfile1.txt")

hitDataset1 <- scoredHits1[[1]]
hitvector1 <- scoredHits1[[2]]

volcano_name <- volcanoPlot(header, hitDataset1, "SigIntensity", "pValue.ttest_1", "GeneName", "Volcano Plot", 0.05, 1)

ZPRIMEQualControl Computing the Z’ factor

Description

Computes the Z’ factor per plate for a complete dataset file and plots the results.

Usage

ZPRIMEQualControl(header, data, channel, plotTitle, showPlot)
**ZScore**

**Arguments**

- **header**: the header of a dataset file generated with `generateDatasetFile`
- **data**: an R data frame generated with `generateDatasetFile`
- **channel**: a character string specifying the name of the column containing the values for computing the $Z'$ factor, e.g. "SigIntensity"
- **plotTitle**: the plot title
- **showPlot**: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

**Value**

Returns the $Z'$ values in the shell for each plate and saves them in a text file. The name of the text file will be the concatenation of the experiment name specified in the header and the character string "Z'Scores.txt".

Shows a plot of the $Z'$ factor values and saves it as a png and a pdf file under the experiment name specified in the header concatenated with the function argument `plotTitle`.

The function returns a list containing:

- **plotName**: the plot name
- **ZPrimeTable**: table containing the $Z'$ values

**References**


**Examples**

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
res <- ZPRIMEQualControl(header, dataset, "SigIntensity", "Z' factors per plate", 1)
zprime_plot <- res[[1]]
zprime_table <- res[[2]]
```

---

**Description**

ZScore normalization (see Value and References)

**Usage**

`ZScore(header, dataset, list0fArgs)`
Arguments

header: the header of a dataset file generated with `generateDatasetFile`
dataset: an R data frame generated with `generateDatasetFile`
listOfArgs: a list containing:
- a character string specifying the column whose values will be used for normalization
- a flag specifying whether controls should be excluded for the computation of the median and median absolute deviation (1) or not (0).

Value

The ZScore is defined as the quotient of the difference between an intensity value and the median of the plate, and of the median absolute deviation.

Returns a list containing:

header: The new header (with an added entry about the normalization procedure in the comments)
dataset: The new dataset with normalized values. The old values are saved in an extra column of the dataset with the suffix ".old"

References


See Also

`ZScorePerScreen`, `BScore`

Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normres <- ZScore(header, dataset, list("SigIntensity", 1))
newheader <- normres[[1]]
newdataset <- normres[[2]]
```

---

ZScore normalization per experiment

Description

ZScore normalization not per plate, but per experiment (see Value and References)

Usage

`ZScorePerScreen(header, dataset, listOfArgs)`
ZScorePlot

Arguments

header the header of a dataset file generated with generateDatasetFile
dataset an R data frame generated with generateDatasetFile
listOfArgs a list containing:
- a character string specifying the column whose values will be used for normalization
- a flag specifying whether controls should be excluded for the computation of the median and median absolute deviation (1) or not (0).

Value

The ZScore is defined as the quotient of the difference between an intensity value and the median of the experiment, and of the median absolute deviation.

Returns a list containing:

header The new header (with an added entry about the normalization procedure in the comments)
dataset The new dataset with normalized values. The old values are saved in an extra column of the dataset with the suffix ".old"

References


See Also

ZScore, BScore

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

normres <- ZScorePerScreen(header, dataset, list("SigIntensity", 1))
newheader <- normres[[1]]
newdataset <- normres[[2]]

ZScorePlot

Plot normalized intensity values per well

Description

Plots the normalized intensity values for each well, together with a black line showing the mean, two green lines showing the standard deviation, and two red lines showing 2 standard deviations.

Usage

ZScorePlot(header, dataset, flag, col4plot, col4anno, plotTitle, showPlot)
Arguments

header: the header of a dataset file generated with `generateDatasetFile`.
dataset: an R data frame generated with `generateDatasetFile`.
flag: either 1 or 2. 1 if the dataset contains values per well, 2 if the dataset contains summarized values for each siRNA (e.g. a dataset summarized with `summarizeReps`).
col4plot: a character string specifying the column whose values will be used for the plot.
col4anno: a character string specifying the column that will be used for the plot annotation.
plotTitle: the plot title.
showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Plots the normalized intensity values for each well, together with a black line showing the mean, and two red lines showing 2 standard deviations. Clicking on the points shows the gene/siRNA name.

The plot is saved as a pdf and a png file named after the experiment name specified in the header concatenated with the `plotTitle`.

The function returns the plot name.

See Also

`plotBar, ZScorePlotTwo`

Examples

```r
data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")
normedvals <- ZScore(header, dataset, list("SigIntensity", 1))
ZScorePlot(normedvals[[1]], normedvals[[2]], 1, "SigIntensity", "GeneName", "Normed intensity values per well", 1)
```

---

### Description

Plots signal intensity values for each well, a black line showing the median, two green lines showing one median absolute deviation, two red lines showing two median absolute deviations.

### Usage

```r
ZScorePlotTwo(header, dataset, flag, flag2, col4plot, col4anno, plotTitle, showPlot)
```
Arguments

header: the header of a dataset file generated with `generateDatasetFile`.
dataset: an R data frame generated with `generateDatasetFile`.
flag: 0, 1, or 2. 0 uses the data from the complete dataset, 1 generates one plot for each experiment, 2 generates one plot for each plate.
flag2: 0 draws lines using mean and sd, 1 draws lines using median and mad.
col4plot: a character string specifying the column whose intensity values will be used for the plot.
col4anno: in case `showPlot` == 1, a character string specifying the column used for identifying points.
plotTitle: the plot title.
showPlot: 0 or 1. 1 will open one or several plot windows in the R GUI, 0 will only save the plot(s) without opening windows.

Value

Saves the plots in pdf and png files named after the experiment name specified in the header concatenated with the `plotTitle` and if applicable the experiment number and/or the plate number.

When `flag` == 0, returns the plot name (`plotName`).

When `flag` == 1, returns a list containing:

- `plotName`: The plot name.
- `minOfScreens`: The number of the first experiment.
- `numOfScreens`: The number of the last experiment.

When `flag` == 2, returns a list containing: the plot name, a vector with the number of the first experiment and of the last experiment, and a vector with the number of the first plate and the number of the last plate.

See Also

`plotBar`, `ZScorePlot`

Examples

data(exampleHeader, package="RNAither")
data(exampleDataset, package="RNAither")

plotname <- ZScorePlotTwo(header, dataset, 0, 1, "SigIntensity", "GeneName", "Data per well", 0)
Index

*Topic IO
  saveDataset, 65
  savepValVec, 66

*Topic arith
  closestToZero, 6
  compareHits, 6
  compareReplicaPlates, 7
  compareReplicates, 8
  compareReplicateSD, 9
  compareReplicateSDPerScreen, 10
  createSubset, 15
  discardLabtek, 16
  discardWells, 17
  divideChannels, 17
  furthestFromZero, 21
  generateReplicateMat, 24
  generateRepMatNoFilter, 25
  hitselectionPval, 27
  hitselectionZscore, 29
  hitselectionZscorePval, 30
  indexSubset, 33
  mainAnalysis, 37
  numCellQualControl, 45
  orderGeneIDs, 46
  percCellQualControl, 47
  replicatesCV, 60
  replicatesSpearmancor, 61
  rms, 62
  rnaither, 62
  SNRQualControl, 67
  summarizeReps, 72
  summarizeRepsNoFiltering, 73
  trim, 74
  ZPRIMEQualControl, 78

*Topic database
  gseaAnalysis, 26
  mainAnalysis, 37
  rnaither, 62

*Topic datagen
  generateDatasetFile, 21
  gseaAnalysis, 26
  joinDatasetFiles, 33
  joinDatasets, 34
  mainAnalysis, 37
  rnaither, 62
  saveOldIntensityColumns, 65
  savepValVec, 66

*Topic datasets
  dataset, 15
  datasetDrosophila, 16
  header, 27
  headerDrosophila, 27
  pValVec1, 57
  pValVec2, 57
  scoredDataset1, 66
  scoredDataset2, 67

*Topic device
  channelPlot, 5
  compareReplicates, 8
  compareReplicateSD, 9
  compareReplicateSDPerScreen, 10
  controlDensity, 11
  controlDensityPerPlate, 12
  controlDensityPerScreen, 13
  DRQualControl, 19
  mainAnalysis, 37
  makeBoxplot4PlateType, 38
  makeBoxplotControls, 39
  makeBoxplotControlsPerPlate, 40
  makeBoxplotControlsPerScreen, 41
  makeBoxplotPerPlate, 42
  makeBoxplotPerScreen, 43
  numCellQualControl, 45
  percCellQualControl, 47
  plotBar, 48
  plotControlHisto, 49
  plotControlHistoPerplate, 50
  plotControlHistoPerscreen, 51
  plotHisto, 52
  plotHistoPerplate, 52
  plotHistoPerscreen, 53
  plotQQ, 54
  plotQQPerplate, 55
  plotQQperscreen, 56
  rnaither, 62
  spatialDistrib, 68
INDEX

spatialDistribHits, 69
vennDiag, 76
volcanoPlot, 77
ZScorePlot, 81
ZScorePlotTwo, 82

*Topic dplot
mainAnalysis, 37
rnaither, 62

*Topic manip
BScore, 4
controlNorm, 14
divideChannels, 17
divNorm, 18
eraseDataSetColumn, 20
incorporatepValVec, 32
LiWongRank, 35
lowessNorm, 36
mainAnalysis, 37
MannWhitney, 44
multTestAdjust, 45
quantileNormalization, 58
rnaither, 62
saveOldIntensityColumns, 65
subtractBackground, 70
sumChannels, 71
varAdjust, 75
ZScore, 79
ZScorePerScreen, 80

*Topic methods
findReplicates, 20
mainAnalysis, 37
rnaither, 62

*Topic misc
mainAnalysis, 37
rnaither, 62

*Topic univar
mainAnalysis, 37
RankProduct, 59
rnaither, 62
Ttest, 74
volcanoPlot, 77

*Topic utilities
mainAnalysis, 37
rnaither, 62

Bscore, 4, 37, 80, 81
channelPlot, 5
closestToZero, 6, 21, 62, 72–74
compareHits, 6
compareReplicaPlates, 7, 8
compareReplicates, 8, 8
compareReplicateSD, 9, 11, 68
compareReplicateSDPerScreen, 9, 10, 68
controlDensity, 11, 12, 13
controlDensityPerPlate, 11, 12, 13
controlDensityPerScreen, 11, 12, 13
controlNorm, 14, 37
createSubset, 15, 33
dataset, 15
datasetDrosophila, 16
discardLabtek, 16, 17
discardWells, 16, 17
divideChannels, 17, 71
divNorm, 18, 37, 57, 66
DRQualControl, 19
eraseDataSetColumn, 20, 71–73
findReplicates, 20
furthestFromZero, 6, 21, 62, 72–74
generateDatasetFile, 4, 5, 7–20, 21, 24, 25,
27–30, 32, 34–37, 39–56, 58–61,
65–73, 75–77, 79–83
generateReplicateMat, 24, 26, 28, 32, 72, 73
generateRepMatNoFilter, 24, 25, 25, 72, 73
gseaAnalysis, 26
header, 27
headerDrosophila, 27
hitselectionPval, 26, 27, 30, 31
hitselectionZscore, 26, 28, 29, 31
hitselectionZscorePval, 28, 30, 30
incorporatepValVec, 32, 45
indexSubset, 15, 33
joinDatasetFiles, 23, 33, 34
joinDatasets, 23, 34, 34
LiWongRank, 35, 37
lowessNorm, 36, 37
mainAnalysis, 37
makeBoxplot4PlateType, 38
makeBoxplotControls, 39, 39, 41–43
makeBoxplotControlsPerPlate, 39, 40, 40,
42, 43
makeBoxplotControlsPerScreen, 39–41, 41,
42, 43
makeBoxplotPerPlate, 39–42, 42, 43
makeBoxplotPerScreen, 39–42, 42, 43
MannWhitney, 7, 28, 30, 37, 44, 57, 59, 67, 75,
77
multTestAdjust, 32, 45
numCellQualControl, 45, 47
order, 46
orderGeneIDs, 46
percCellQualControl, 46, 47
plotBar, 48, 82, 83
plotControlHisto, 49, 50, 51
plotControlHistoPerplate, 49, 50, 51
plotControlHistoPerscreen, 49, 50, 51
plotHisto, 52, 53, 54
plotHistoPerplate, 52, 52, 54
plotHistoPerscreen, 52, 53, 53
plotQQ, 54, 56, 57
plotQQperplate, 55, 55, 57
plotQQperscreen, 55, 56, 56
pValVec1, 57
pValVec2, 57
quantileNormalization, 37, 58
RankProduct, 28, 30, 37, 44, 59, 75
replicatesCV, 60
replicatesSpearmanCor, 61
rms, 6, 21, 62, 72–74
RNAither (RNAither-package), 3
rnaither, 62
RNAither-package, 3
saveDataset, 65
saveOldIntensityColumns, 65, 71
savePValVec, 66
scoredDataset1, 66
scoredDataset2, 67
SNRQualControl, 67
spatialDistrib, 9–11, 68
spatialDistribHits, 69
subtractBackground, 37, 70
sumChannels, 18, 71
summarizeReps, 6, 21, 62, 72, 73, 74, 82
summarizeRepsNoFiltering, 6, 21, 62, 72, 73, 74
trim, 6, 21, 62, 72, 73, 74
Ttest, 7, 26, 28, 30–32, 37, 44, 45, 57, 59, 66, 70, 74, 77, 78
varAdjust, 37, 75
vennDiag, 7, 76
volcanoPlot, 77
ZPRIMEQualControl, 78
ZScore, 37, 79, 81
ZScorePerScreen, 37, 80, 80
ZScorePlot, 48, 81, 83
ZScorePlotTwo, 48, 82, 82