Package 'highSCREEN'

August 22, 2018

Type Package

Title High-Throughput Screening for Plate Based Assays

Version 0.3

Description Can be used to carry out extraction, normalization, quality control (QC), candidate hits identification and visualization for plate based assays, in drug discovery. This project was funded by the Division of Allergy, Immunology, and Transplantation, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under contract No. HHSN272201400054C entitled ``Adjuvant Discovery For Vaccines Against West Nile Virus and Influenza'', awarded to Duke University and lead by Drs. Herman Staats and Soman Abraham.

Depends R (>= 3.4.0), gplots (>= 3.0.1)

VignetteBuilder knitr

Suggests knitr

License GPL-3

LazyLoad yes

NeedsCompilation no

Author Ivo D. Shterev [aut, cre],
Cliburn Chan [aut],
Gregory D. Sempowski [aut]

Maintainer Ivo D. Shterev <i.shterev@duke.edu>

Repository CRAN

Date/Publication 2018-08-22 15:10:02 UTC

R topics documented:

highSCREEN-package *High-Throughput Screening for Plate Based Assays*

Description

This package can be used to carry out extraction, normalization, quality control (QC), candidate hits identification and visualization for plate based assays, in drug discovery.

Details

The DESCRIPTION file:

Package:	highSCREEN
Type:	Package
Title:	High-Throughput Screening for Plate Based Assays
Version:	0.3
Authors@R:	c(person(c("Ivo", "D."), "Shterev", role = c("aut", "cre"), email = "i.shterev@duke.edu"), person("Cliburn"
Description:	Can be used to carry out extraction, normalization, quality control (QC), candidate hits identification and v
Depends:	R (>= 3.4.0), gplots (>= 3.0.1)
VignetteBuilder:	knitr
Suggests:	knitr
License:	GPL-3
LazyLoad:	yes
Author:	Ivo D. Shterev [aut, cre], Cliburn Chan [aut], Gregory D. Sempowski [aut]
Maintainer:	Ivo D. Shterev <i.shterev@duke.edu>

Index of help topics:

```
extractplate          Extract plate
formatRESULT          Format results
highSCREEN-package    High-Throughput Screening for Plate Based
                      Assays
hits                  Identify hits
normplate             Normalize plate
plotcont              Plot control density
plotplate             Plot plate activity
qcplate               Quality control
rankhits              Rank hits
zfactor.ssmd          Compute Z-factor and SSMD
```

Author(s)

I. D. Shterev, C. Chan and G. D. Sempowski

Maintainer: I. D. Shterev <i.shterev@duke.edu>

---

extractplate	*Extract plate*

Description

description

Usage

```
extractplate(dat0, dat1, plate, replicate)
```

Arguments

dat0	A list consisting of the following elements for the t0-specific data set: $Replicate1$: Data frame containing the first triplicate and 4 additional columns specifying the controls and their plate locations. $Replicate2$: Data frame containing the second triplicate and 4 additional columns specifying the controls and their plate locations. $Replicate3$: Data frame containing the third triplicate and 4 additional columns specifying the controls and their plate locations.
dat1	A list consisting of the following elements for the t1-specific data set: $Replicate1$: Data frame containing the first triplicate and 4 additional columns specifying the controls and their plate locations. $Replicate2$: Data frame containing the second triplicate and 4 additional columns specifying the controls and their plate locations. $Replicate3$: Data frame containing the third triplicate and 4 additional columns specifying the controls and their plate locations.
plate	Specifies which of the 4 plates to be extracted.
replicate	Specfies which replicate to be extracted.

Value

This function returns a list consisting of the following elements:

dat0	Data frame containing the t0-specific plate of compounds and controls.
dat1	Data frame containing the t1-specific plate of compounds and controls.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicate to create t0-specific data set
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicate to create t1-specific data set
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 3, replicate 2
extractplate(replicates_t0, replicates_t1, plate=3, replicate=2)
```

---

formatRESULT   *Format results*

Description

description

Usage

```
formatRESULT(dat, replicate="Replicate", score="S", t="Time")
```

Arguments

dat	Data frame such as the return data frame of the function $normplate()$.
replicate	Specifies the column name of the input data frame that contains the replicate numbers.

score	Specifies the column name of the input data frame that contains the normalized score values.
t	Specifies the column name of the input data frame that contains the time designations (0 or 1).

Value

This function returns a data frame with each row corresponding to a compound/control and the following columns:

ID	Has the complete information to identify a compound or control. It contains information about the main plate, the quadrant/plate and the well.
MainPlate	Specifies the main plate to wich the compound/control belongs.
Plate	Specifies the quadrant/plate to wich the compound/control belongs.
Norm	Specifies the normalization method that was applied for the specific compound.
well	Specifies the location (row and column) of the compound/control in the quadrant.
row	Specifies the row location in the quadrant.
col	Specifies the column location in the quadrant.
welltype	Specifies if the well is compound or control.
S0	Replicates of the score from the t0-specific data set.
S1	Replicates of the score from the t1-specific data set.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t0-specific data
```

```
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 1, all replicates
dat1 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)
dat2 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)
dat3 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

# normalize data of all replicates
res1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
plate=1, replicate=1, norm="zscore")
res2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
plate=1, replicate=2, norm="zscore")
res3 = normplate("Main Plate 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
plate=1, replicate=3, norm="zscore")

# reformat data of all replicates
head(formatRESULT(rbind(res1, res2, res3), replicate="Replicate", t="Time"))
```

---

hits	*Identify hits*

Description

description

Usage

```
hits(dat.raw, dat.norm, s0="S0", s1="S1",
qc.mainplates, qc1.val=0.225, hit.val=3)
```

Arguments

dat.raw	Data frame containing raw data as an output from $formatRESULT()$.
dat.norm	Data frame containing normalized data set as an output from $formatRESULT()$.
s0	Specifies the name of the columns containing t0-specific scores.

s1	Specifies the name of the columns containing t1-specific scores.
qc.mainplates	A vector containing names of main plates that passed QC.
qc1.val	Threshold value for QC1.
hit.val	Threshold value for identifying candidate hits, based on the mean of t1-specific scores.

Value

The function returns a data frame. Each row corresponds to a compound that passed QC1 and belongs to a plate that passed overall QC. The data frame contains the following columns:

ID	Has the complete information to identify a compound or control. It contains information about the main plate, the quadrant/plate and the well.
MainPlate	Specifies the main plate to wich the compound/control belongs.
Plate	Specifies the quadrant/plate to wich the compound/control belongs.
Norm	Specifies the normalization method that was applied for the specific compound.
well	Specifies the location (row and column) of the compound/control in the quadrant.
row	Specifies the row location in the quadrant.
col	Specifies the column location in the quadrant.
welltype	Specifies if the well is compound or control.
S0	Replicates of the score from the t0-specific data set.
S1	Replicates of the score from the t1-specific data set.
IND2	Indicator variable specifying if the compound passes ($TRUE$) or fails ($FALSE$) QC2.
IND3	Indicator variable specifying if the compound passes ($TRUE$) or fails ($FALSE$) QC3.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]
```

```
# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# combine all replicates for the t0-specific data
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 1, replicate 1
dat1 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)

# extract plate 1, replicate 2
dat2 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)

# extract plate 1, replicate 3
dat3 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

# no normalizion
datraw1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
 plate=1, replicate=1, norm="raw")
datraw2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
 plate=1, replicate=2, norm="raw")
datraw3 = normplate("Main Pltae 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
 plate=1, replicate=3, norm="raw")

# combine 3 replicate
datraw = rbind(datraw1, datraw2, datraw3)

# reformat result
datraw = formatRESULT(datraw, replicate="Replicate", t="Time")

# c-score normalization
datnorm1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
 plate=1, replicate=1, norm="cscore",
poscont="Control P", negcont="Control N")
datnorm2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
 plate=1, replicate=2, norm="cscore",
poscont="Control P", negcont="Control N")
```

```
datnorm3 = normplate("Main Pltae 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
 plate=1, replicate=3, norm="cscore",
poscont="Control P", negcont="Control N")

# combine 3 replicates
datnorm = rbind(datnorm1, datnorm2, datnorm3)

# reformat result
datnorm = formatRESULT(datnorm, replicate="Replicate", t="Time")

# identify hits
head(hits(datraw, datnorm, qc.mainplates="Main Plate 1", qc1.val=0.225, hit.val=3))
```

---

normplate   *Normalize plate*

Description

description

Usage

```
normplate(mainplate, dat0, dat1, cmap, plate, replicate, norm="bscore",
 poscont=NULL, negcont=NULL)
```

Arguments

mainplate	Main Plate designation.
dat0	Data frame containing the 96-well t0-specific plate/quadrant to be normalized.
dat1	Data frame containing the 96-well t1-specific plate/quadrant to be normalized.
cmap	96-well plate control map.
plate	Plate/quadrant designation. Ranges from 1 to 4.
replicate	Replicate designation.
norm	Normalization method to be applied. Currenlty implemented methods include C-score (*cscore*), B-score (*bscore*), Z-score (eqnzscore) and no normalization (*raw*).
poscont	Specifies positive control designation (used in C-score normalization).
negcont	Specifies negative control designation (used in C-score normalization).

Value

Data frame containing the following columns:

MainPlate	Specifies the main plate to wich the compound/control belongs.
Time	Specifies t0 (0) or t1 (1) specific data.
Plate	Specifies the quadrant/plate to wich the compound/control belongs.

Replicate	Specifies replicate.
Norm	Specifies the normalization method that was applied for the specific compound.
well	Specifies the location (row and column) of the compound/control in the quadrant.
row	Specifies the row location in the quadrant.
col	Specifies the column location in the quadrant.
score	Shows the score value.
welltype	Specifies if the well is compound or control.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]
cmap

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicate for the t0-specific data
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicate for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")
```

```
# extract plate 1, replicate 1
dat = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)

# normalize using c-score
head(normplate("Main Plate 1", dat[["dat0"]], dat[["dat1"]], cmap,
plate=1, replicate=1, norm="cscore",
 poscont="Control P", negcont="Control N"))

# normalize using b-score (medpolish)
head(normplate("Main Plate 1", dat[["dat0"]], dat[["dat1"]], cmap,
plate=1, replicate=1, norm="bscore"))

# normalize using z-score
head(normplate("Main Plate 1", dat[["dat0"]], dat[["dat1"]], cmap,
plate=1, replicate=1, norm="zscore"))
```

---

plotcont	*Plot control density*

Description

description

Usage

```
plotcont(dat, score="S", main, xaxis.marks=seq(0,5,0.025))
```

Arguments

dat	Data frame as an output from $normplate()$.
score	Specifies the columns containing the data.
main	Main title for the plot.
xaxis.marks	Specifies x-axis mark ticks.

Examples

```
set.seed(1234)

nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
```

```
    floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
    rep(c("Control low", "Control med", "Control high"),
    (floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
    cmap = cmap[seq(1,nr,2),]

    # create 2nd replicate of data matrix with compounds and controls
    r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

    # create 3rd replicate of data matrix with compounds and controls
    r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

    # combine all replicates for the t0-specific data
    replicates_t0 = list(r1, r2, r3)
    names(replicates_t0) = c("R1", "R2", "R3")

    # create 1st replicate of data matrix with compounds and controls
    r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

    # create 2nd replicate of data matrix with compounds and controls
    r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

    # create 3rd replicate of data matrix with compounds and controls
    r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

    # combine all replicates for the t1-specific data
    replicates_t1 = list(r1, r2, r3)
    names(replicates_t1) = c("R1", "R2", "R3")

    # extract plate 1, replicate 1
    dat11 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)
    # extract plate 1, replicate 2
    dat12 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)
    # extract plate 1, replicate 3
    dat13 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

    # no normalizion (norm="raw")
    res11 = normplate("Main Plate 1", dat11[["dat0"]], dat11[["dat1"]], cmap,
     plate=1, replicate=1, norm="raw")
    res12 = normplate("Main Plate 1", dat12[["dat0"]], dat12[["dat1"]], cmap,
     plate=1, replicate=2, norm="raw")
    res13 = normplate("Main Pltae 1", dat13[["dat0"]], dat13[["dat1"]], cmap,
     plate=1, replicate=3, norm="raw")

    # combine 3 replicates
    res1 = rbind(res11, res12, res13)
    # reformat result
    res1 = formatRESULT(res1, replicate="Replicate", t="Time")

    layout(matrix(c(1,2,3), 3, 1, byrow = TRUE))

    # plot density of all positive controls
    plotcont(subset(res1, welltype=="Control P"), main="Density of Positive Controls",
     xaxis.marks=seq(-1,5,0.025))
```

```
# plot density of all negative controls
plotcont(subset(res1, welltype=="Control N"), main="Density of Negative Controls",
 xaxis.marks=seq(-1,5,0.025))

# plot density of controls with low, medium and high concentrations
plotcont(subset(res1, welltype=="Control low" | welltype=="Control med" |
welltype=="Control high"), main="Density of Controls with Low,
Medium and High Concentrations", xaxis.marks=seq(-1,5,0.025))
```

---

plotplate	*Plot plate activity*

Description

description

Usage

```
plotplate(dat, score="S", main)
```

Arguments

dat	Data frame as an output from $normplate()$.
score	Specifies the columns containing the data.
main	Main title for the plot.

Examples

```
set.seed(1234)

nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)
```

```
# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t0-specific data
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicates of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 1, replicate 1
dat11 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)
# extract plate 1, replicate 2
dat12 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)
# extract plate 1, replicate 3
dat13 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

# no normalizion (norm="raw")
res11 = normplate("Main Plate 1", dat11[["dat0"]], dat11[["dat1"]], cmap,
 plate=1, replicate=1, norm="raw")
res12 = normplate("Main Plate 1", dat12[["dat0"]], dat12[["dat1"]], cmap,
 plate=1, replicate=2, norm="raw")
res13 = normplate("Main Pltae 1", dat13[["dat0"]], dat13[["dat1"]], cmap,
 plate=1, replicate=3, norm="raw")

# combine 3 replicates
res1 = rbind(res11, res12, res13)
# reformat result
res1 = formatRESULT(res1, replicate="Replicate", t="Time")

# plot single plate activity levels
plotplate(res1, main="Single Plate Activity Levels")
```

---

qcplate   *Quality control*

Description

description

Usage

```
qcplate(dat, s0="S0", s1="S1", poscont="Control P",
 negcont="Control N", qc1.val=0.225, qc2.val=2, addcont, welltype="welltype")
```

Arguments

`dat`	
`s0`	Specifies the name of the columns containing t0-specific scores.
`s1`	Specifies the name of the columns containing t1-specific scores.
`poscont`	Specifies positive control designation (used in C-score normalization).
`negcont`	Specifies negative control designation (used in C-score normalization).
`qc1.val`	Pre-defined threshold value for QC1.
`qc2.val`	Pre-defined threshold value for QC2.
`addcont`	A vector character names specifying the additional controls for QC3. The vignette provides more details on the particular order of specifying the additional controls.
`welltype`	Specifies the name for the column defining the well type.

Details

details

Value

value

References

references

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]
```

```
# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t0-specific data
replicates_t0 = list(r1, r2, r3)

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 1, replicate 1
dat11 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)

# extract plate 1, replicate 2
dat12 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)

# extract plate 1, replicate 3
dat13 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

# no normalizion (norm="raw")
res11 = normplate("Main Plate 1", dat11[["dat0"]], dat11[["dat1"]], cmap,
 plate=1, replicate=1, norm="raw")
res12 = normplate("Main Plate 1", dat12[["dat0"]], dat12[["dat1"]], cmap,
 plate=1, replicate=2, norm="raw")
res13 = normplate("Main Pltae 1", dat13[["dat0"]], dat13[["dat1"]], cmap,
 plate=1, replicate=3, norm="raw")

# combine 3 replicates
res1 = rbind(res11, res12, res13)

# reformat result
res1 = formatRESULT(res1, replicate="Replicate", t="Time")

# perform QC
qcplate(res1, poscont="Control P", negcont="Control N", qc1.val=0.225, qc2.val=2,
addcont=c("Control low", "Control med", "Control high"), welltype="welltype")
```

---

rankhits   *Rank hits*

Description

This function ranks the candidate compound hits based on the mean of t1-specific scores.

Usage

```
rankhits(dat, score.before="S0", score.after="S1", var="m1")
```

Arguments

dat	Data frane as the output of the function $formatRESULT$.
score.before	Specifies the name of the columns containing t0-specific scores.
score.after	Specifies the name of the columns containing t1-specific scores.
var	Specifies the variable according to which the ranking is done.

Value

Data frame with a row for each hit. The columns are as follows:

ID	Has the complete information to identify a compound or control. It contains information about the main plate, the quadrant/plate and the well.
MainPlate	Specifies the main plate to wich the compound/control belongs.
Plate	Specifies the quadrant/plate to wich the compound/control belongs.
Norm	Specifies the normalization method that was applied for the specific compound.
well	Specifies the location (row and column) of the compound/control in the quadrant.
row	Specifies the row location in the quadrant.
col	Specifies the column location in the quadrant.
welltype	Specifies if the well is compound or control.
S0	Replicates of the score from the t0-specific data set.
S1	Replicates of the score from the t1-specific data set.
IND2	Indicator variable specifying if the compound passes ($TRUE$) or fails ($FALSE$) QC2.
IND3	Indicator variable specifying if the compound passes ($TRUE$) or fails ($FALSE$) QC3.
diff	Difference between mean of t1 and t0 replicates.
m0	Mean of t0-specific replicates (MB).
s0	Standard deviation of t0-specific replicates (SB).
rs0	Coefficient of variation of t0-specific replicates. It is the ratio of S0 and M0.
m1	Mean of t1-specific replicates (MA).
s1	Standard deviation of t1-specific replicates (SA)
rs1	Coefficient of variation of t1-specific replicates. It is the ratio of S1 and M1.
ind_below	see ind below
ind_above	see ind below
ind	$TRUE$ (if both ind_below and ind_above are $TRUE$) if RS1 is within the $1.5 \times IQR$, where interquartile range (IQR) is computed based on all hits.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)*0.01), nr, nc)

# combine all replicates for the t0-specific data
replicates_t0 = list(r1, r2, r3)
names(replicates_t0) = c("R1", "R2", "R3")

# create 1st replicate of data matrix with compounds and controls
r1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
r2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
r3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t1-specific data
replicates_t1 = list(r1, r2, r3)
names(replicates_t1) = c("R1", "R2", "R3")

# extract plate 1, replicate 1
dat1 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=1)
# extract plate 1, replicate 2
dat2 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=2)
# extract plate 1, replicate 3
dat3 = extractplate(replicates_t0, replicates_t1, plate=1, replicate=3)

# no normalizion
datraw1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
 plate=1, replicate=1, norm="raw")
datraw2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
 plate=1, replicate=2, norm="raw")
```

```
datraw3 = normplate("Main Pltae 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
 plate=1, replicate=3, norm="raw")

# combine 3 replicates
datraw = rbind(datraw1, datraw2, datraw3)

# reformat result
datraw = formatRESULT(datraw, replicate="Replicate", t="Time")

# c-score normalization
datnorm1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
 plate=1, replicate=1, norm="cscore",
poscont="Control P", negcont="Control N")
datnorm2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
 plate=1, replicate=2, norm="cscore",
poscont="Control P", negcont="Control N")
datnorm3 = normplate("Main Pltae 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
 plate=1, replicate=3, norm="cscore",
poscont="Control P", negcont="Control N")

# combine 3 replicates
datnorm = rbind(datnorm1, datnorm2, datnorm3)

# reformat result
datnorm = formatRESULT(datnorm, replicate="Replicate", t="Time")

# identify hits
h = hits(datraw, datnorm, qc.mainplates="Main Plate 1", qc1.val=0.225, hit.val=3)

# rank hits in disending order of mean of t1-specific replicate scores "m1"
head(rankhits(h))
```

---

zfactor.ssmd   *Compute Z-factor and SSMD*

Description

This function computes the Z-factor and strictly standardized mean difference (SSMD) of a given 96-well plate.

Usage

```
zfactor.ssmd(dat, pos.cont, neg.cont, MainPlate, replicate)
```

Arguments

dat	Data frame as an output of the function $formatRESULT()$.
pos.cont	Designation of positive control.
neg.cont	Designation of negative control.

MainPlate   Specifies main plate.

replicate   Specifies the replicate.

Value

Returns a data frame with one row and the following columns:

MainPlate   Specifies the main plate.

replicate   Specifies the replicate.

ZFactor_Before   Specifies the Z-factor computed based on the t0-specific data.

ZFactor_After   Specifies the Z-factor computed based on the t1-specific data.

SSMD_Before   Specifies SSMD computed based on the t0-specific data.

SSMD_After   Specifies SSMD computed based on the t1-specific data.

Examples

```
set.seed(1234)
nc = 24
nr = 16

# create 1st replicate of data matrix with compounds and controls
replicate1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create control map
cmap = data.frame(X1=c(rep("Control P", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control N",
floor(nr/3))), X2=c(rep("Control N", floor(nr/3)),
rep(c("Control low", "Control med", "Control high"),
(floor(nr/3)+nr-3*floor(nr/3))/3), rep("Control P", floor(nr/3))))
cmap = cmap[seq(1,nr,2),]

# create 2nd replicate of data matrix with compounds and controls
replicate2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
replicate3 = matrix(abs(rnorm(nr*nc)), nr, nc)

# combine all replicates for the t0-specific data
replicates_before = list(replicate1, replicate2, replicate3)
names(replicates_before) = c("Replicate1", "Replicate2", "Replicate3")

# create 1st replicate of data matrix with compounds and controls
replicate1 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 2nd replicate of data matrix with compounds and controls
replicate2 = matrix(abs(rnorm(nr*nc)), nr, nc)

# create 3rd replicate of data matrix with compounds and controls
replicate3 = matrix(abs(rnorm(nr*nc)), nr, nc)
```

```
# combine all replicates for the t1-specific data
replicates_after = list(replicate1, replicate2, replicate3)
names(replicates_after) = c("Replicate1", "Replicate2", "Replicate3")

# extract plate 1, replicate 1
dat1 = extractplate(replicates_before, replicates_after, plate=1, replicate=1)
# extract plate 1, replicate 2
dat2 = extractplate(replicates_before, replicates_after, plate=1, replicate=2)
# extract plate 1, replicate 3
dat3 = extractplate(replicates_before, replicates_after, plate=1, replicate=3)

# no normalizion
datraw1 = normplate("Main Plate 1", dat1[["dat0"]], dat1[["dat1"]], cmap,
 plate=1, replicate=1, norm="raw")
datraw2 = normplate("Main Plate 1", dat2[["dat0"]], dat2[["dat1"]], cmap,
 plate=1, replicate=2, norm="raw")
datraw3 = normplate("Main Pltae 1", dat3[["dat0"]], dat3[["dat1"]], cmap,
 plate=1, replicate=3, norm="raw")

# combine 3 replicates
datraw = rbind(datraw1, datraw2, datraw3)

# reformat result
datraw = formatRESULT(datraw, replicate="Replicate", t="Time")

# compute z-factor and ssmd for each raw compound, replicate 1
zfactor.ssmd(datraw, "Control P", "Control N", "Main Plate 1", 1)
```

Index