Package 'displayHTS'

February 19, 2015

Type Package

Title displayHTS

Version 1.0

Date 2013-01-07

Author Xiaohua Douglas Zhang & Zhaozhi Zhang

Maintainer Zhaozhi Zhang <zhang.zhaozhi7@gmail.com>

Description A package containing R functions for displaying data and results from high-throughput screening experiments.

License GPL (>= 2)

Repository CRAN

Date/Publication 2013-01-07 15:57:30

NeedsCompilation no

R topics documented:

displayHTS-package *Display high-throughput screening data and results*

Description

An R package that displays data and results from high-throughput screening experiments.

Details

Package:	displayHTS
Type:	Package
Version:	1.0
Date:	2013-01-07
License:	GPL(>=2)

The R package displayHTS has four main functions for generating graphics to display data and results from HTS experiments: `plateWellSeries.fn`, `imageDesign.fn`, `imageIntensity.fn`, and `dualFlashlight.fn`. `plateWellSeries.fn` generates a scatter plot based on the measured or calculated values of each well in every plate in a HTS experiment. `imageDesign.fn` displays a plate design that can be used to both visualize the arrangement of controls and samples in a plate. `imageIntensity.fn` creates an image plot that shows the intensities or calculated values of every well utilizing the boxplot statistics and allows for easy analysis of any systematic errors of measurement; finally, `dualFlashlight.fn` generates the dual-flashlight plot, volcano plot and plate correlation plot.

This package also includes three example datasets: `HTSdata`, `HTSdataSort` and `HTSresults`. `HTSdata` contain the raw data; after sorting and removing redundant records, the data are stored into `HTSdataSort`. Finally, the data are processed and the SSMD, mean, p-value, and number of replicates of the data are contained in the dataset `HTSresults`.

Author(s)

Xiaohua Douglas Zhang & Zhaozhi Zhang

Maintainer: Zhaozhi Zhang <zhang.zhaozhi7@gmail.com>

References

Zhang XHD, Zhang ZZ, 2013. displayHTS: a R package displaying data and results from high-throughput screening experiments. Bioinformatics (submitted).

Zhang XHD, 2011. Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-scale RNAi Research. Cambridge University Press, Cambridge, UK.

Zhang XHD, Santini F, Lacson R, Marine SD, Wu Q, Benetti L, Yang R, McCampbell A, Berger JP, Toolan DM, Stec EM, Holder DJ, Soper KA, Heyse JF and Ferrer M. 2011. cSSMD: Assessing collective activity of multiple siRNAs in genome-scale RNAi screens. Bioinformatics 27(20): 2775-2781.

Zhang XHD. 2011. Illustration of SSMD, z-score, SSMD*, z*-score and t-statistic for hit selection in high-throughput screens. Journal of Biomolecular Screening 16 (7): 775 - 785.

Zhang XHD. 2010. Assessing the size of gene or RNAi effects in multi-factor high-throughput experiments. Pharmacogenomics 11(2): 199 - 213.

Zhang XHD. 2009. A method effectively comparing gene effects in multiple conditions in RNAi and expression profiling research. Pharmacogenomics 10(3):345-358.

Zhang XHD, Espeseth AS, Johnson E, Chin J, Gates A, Mitnaul L, Marine SD, Tian J, Stec EM, Kunapuli P, Holder DJ, Heyse JF, Stulovici B, Ferrer M. 2008. Integrating experimental and analytic approaches to improve data quality in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 378-389.

Zhang XHD, 2008. Novel analytic criteria and effective plate designs for quality control in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 363-377.

Zhang XHD. 2007. A new method with flexible and balanced control of false negatives and false positives for hit selection in RNA interference high throughput screening assays. Journal of Biomolecular Screening 12 (5): 645-655.

Zhang XHD, Ferrer M, Espeseth AS, Marine SD, Stec EM, Crackower MA, Holder DJ, Heyse JF, Strulovici B. 2007. The use of strictly standardized mean difference for hit selection in primary RNA interference high throughput screening experiments. Journal of Biomolecular Screening 12 (4): 497-509.

Zhang XHD. 2007. A pair of new statistical parameters for quality control in RNA interference high throughput screening assays. Genomics 39: 552-561.

Zhang XHD, Yang XC, Chung N, Gates AT, Stec EM, Kunapuli P, Holder DJ, Ferrer M, Espeseth AS. 2006. Robust statistical methods for hit selection in RNA interference high throughput screening experiments. Pharmacogenomics 7 (3) 299-309.

Examples

```
#########################################################################
## generate the figures in the article where this package is published
#########################################################################
# fig1.A: plate-well series plot
data(HTSdataSort)
wells = as.character(unique(HTSdataSort[, "WELL_USAGE"]))
colors = c("black",  "pink", "grey", "blue", "skyblue", "green", "red")
orders=c(1, 3, 2, 4, 5, 7, 6)
par( mfrow=c(1,1) )
plateWellSeries.fn(data.df = HTSdataSort[1:(384*2),],
                   intensityName="log2Intensity",
                   plateName="BARCODE", wellName="WELL_USAGE",
                   rowName="XPOS", colName="YPOS", show.wellTypes=wells,
                   order.wellTypes=orders, color.wells=colors,
                   pch.wells=rep(1, 7), ppf=6, byRow=TRUE,
                   yRange=NULL, cex.point=0.75,cex.legend=0.75,
                   main="A: Plate-well series plot")

# fig1.B: hit and control image
data(HTSresults)
condtSample = HTSresults[, "WELL_USAGE"] == "Sample"
```

```
condtUp = HTSresults[,"ssmd"] >= 1 & HTSresults[,"mean"] >= log2(1.2)
condtDown = HTSresults[,"ssmd"] <= -1 & HTSresults[,"mean"] <= -log2(1.2)
sum(condtSample & (condtUp | condtDown) )/sum(condtSample)
hit.vec = as.character(HTSresults[, "WELL_USAGE"])
hit.vec[ condtSample & condtUp ] = "up-hit"
hit.vec[ condtSample & condtDown ] = "down-hit"
hit.vec[ condtSample & !condtUp & !condtDown] = "non-hit"
result.df = cbind(HTSresults, "hitResult"=hit.vec)
wells = as.character(unique(result.df[, "hitResult"])); wells
colors = c("black",  "green", "white", "red",
           "grey", "purple1", "purple2", "pink", "purple3")
par( mfrow=c(1,1) )
imageDesign.fn(result.df[1:384,], wellName="hitResult", rowName="XPOS",
                colName="YPOS", wells=wells, colors=colors,
                title="B: Image of hits and controls")

## fig1.C: dual-flashlight plot,
par( mfrow=c(1, 1) )
dualFlashlight.fn(HTSresults, wellName="WELL_USAGE", x.name="mean",
                  y.name="ssmd", sampleName="Sample", sampleColor="black",
                  controls = c("negCTRL", "posCTRL1", "mock1"),
                  controlColors = c("green", "red", "lightblue"),
                  xlab="Average Fold Change", ylab="SSMD",
                  main="C: Dual-Flashlight Plot", x.legend=0.1, y.legend= -12,
                  cex.point=1, cex.legend = 0.8,
                  xat=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
                  xMark=c("1/4", "1/2", "1/1.2","1", "1.2", "2", "4"),
                  xLines=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
                  yLines=c(-5, -3, -2, -1, 0, 1, 2, 3, 5 ) )

## fig1.D: volcano plot,
result.df = cbind(HTSresults, "neg.log10.pval" = -log10(HTSresults[,"p.value"]))
dualFlashlight.fn(result.df, wellName="WELL_USAGE", x.name="mean",
                  y.name="neg.log10.pval",
                  sampleName="Sample", sampleColor="black",
                  controls = c("negCTRL", "posCTRL1", "mock1"),
                  controlColors = c("green", "red", "lightblue"),
                  xlab="Average Fold Change", ylab="p-value in -log10 scale",
                  main="D: Volcano Plot", x.legend=NA, y.legend=-log10(0.006),
                  cex.point=1, cex.legend = 0.8,
                  xat=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
                  xMark=c("1/4", "1/2", "1/1.2","1", "1.2", "2", "4"),
                  xLines=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
                  yLines=c(-5, -3, -2, -1, 0, 1, 2, 3, 5 ) )
```

---

dualFlashlight.fn *Dual-Flashlight Plot*

Description

A function that draws a dual-flashlight plot, volcano plot, and plate correlation plot.

Usage

```
dualFlashlight.fn(result.df, wellName = "WELL_USAGE", x.name = "mean", y.name = "ssmd", sampleName = '
```

Arguments

result.df	A data frame containing data or results from an HTS experiment. it must includes at least three columns for well name, x (e.g., average fold change in log) and y (e.g.,SSMD), respectively.
wellName	Name of the column in result.df indicating well names.
x.name	Name of the column in result.df indicating values for x-axis.
y.name	Name of the column in result.df indicating values for y-axis.
sampleName	The name of the well type indicating sample wells.
sampleColor	The color for sample wells.
controls	A vector including controls to be shown in the plot.
controlColors	A vector including the color of the controls to be shown in the plot. It must have the same length as 'controls'.
xlab	Same as internal function 'plot'.
ylab	Same as internal function 'plot'.
main	Same as internal function 'plot'.
x.legend	The x-coordinates to be used to position the legend.
y.legend	The y-coordinates to be used to position the legend.
cex.point	Defines the size of points in the plot.
cex.legend	Defines the size of the legend in the plot.
xat	The position of x-axis to be labeled.
xMark	The labels of x-axis corresponding to 'xat'.
yat	The position of y-axis to be labeled.
yMark	The labels of y-axis corresponding to 'yat'.
xLines	X-values indicating positions of vertical grey lines to be drawn.
yLines	Y-values indicating positions of horizontal grey lines to be drawn.

Author(s)

Xiaohua Douglas Zhang & Zhaozhi Zhang

References

Zhang XHD, Zhang ZZ. 2013. displayHTS: a R package displaying data and results from high-throughput screening experiments. Bioinformatics (submitted).

Zhang XHD, 2011. Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-scale RNAi Research. Cambridge University Press, Cambridge, UK.

Zhang XHD. 2009. A method effectively comparing gene effects in multiple conditions in RNAi and expression profiling research. Pharmacogenomics 10(3):345-358.

Zhang XHD. 2010. Assessing the size of gene or RNAi effects in multi-factor high-throughput experiments. Pharmacogenomics 11(2): 199 - 213.

Zhang XHD, Santini F, Lacson R, Marine SD, Wu Q, Benetti L, Yang R, McCampbell A, Berger JP, Toolan DM, Stec EM, Holder DJ, Soper KA, Heyse JF and Ferrer M. 2011. cSSMD: Assessing collective activity of multiple siRNAs in genome-scale RNAi screens. Bioinformatics 27(20): 2775-2781.

See Also

plateWellSeries.fn, imageDesign.fn, imageIntensity.fn

Examples

```
# for dual-flashlight plot
data("HTSresults", package = "displayHTS")
par( mfrow=c(1, 1) )
dualFlashlight.fn(HTSresults, wellName="WELL_USAGE", x.name="mean",
                  y.name="ssmd", sampleName="Sample", sampleColor="black",
                  controls = c("negCTRL", "posCTRL1", "mock1"),
                  controlColors = c("green", "red", "lightblue"),
                  xlab="Average Fold Change", ylab="SSMD",
                  main="Dual-Flashlight Plot", x.legend=NA, y.legend=NA,
                  cex.point=1, cex.legend = 0.8,
                  xat=log2( c(1/8, 1/4, 1/2, 1, 2, 4, 8) ),
                  xMark=c("1/8", "1/4", "1/2", "1", "2", "4", "8"),
                  xLines=log2(c(1/4, 1/2 ,1, 2, 4)),
                  yLines=c(-5, -3, -2, -1, 0, 1, 2, 3, 5 ) )
# for volcano plot
result.df=cbind(HTSresults,"neg.log10.pval"=-log10(HTSresults[,"p.value"]))
dualFlashlight.fn(result.df, wellName="WELL_USAGE", x.name="mean",
                  y.name="neg.log10.pval",
                  sampleName="Sample", sampleColor="black",
                  controls = c("negCTRL", "posCTRL1", "mock1"),
                  controlColors = c("green", "red", "lightblue"),
                  xlab="Average Fold Change", ylab="p-value",
                  main="Volcano Plot", x.legend=NA, y.legend=-log10(0.06),
                  cex.point=1, cex.legend = 0.8,
                  xat=log2( c(1/8, 1/4, 1/2, 1, 2, 4, 8) ),
                  xMark=c("1/8", "1/4", "1/2", "1", "2", "4", "8"),
                  yat=-log10( c(0.00001, 0.0001, 0.001, 0.01, 0.1, 1) ),
                  yMark=c(0.00001, 0.0001, 0.001, 0.01, 0.1, 1),
                  xLines=log2(c(1/4, 1/2 ,1, 2, 4)),
                  yLines=-log10( c( 0.001, 0.01, 0.05) ) )
```

```
# plate pair correlation plot
data("HTSdataSort", package = "displayHTS")
data.df= cbind(HTSdataSort[1:384,], HTSdataSort[384+1:384,])
names(data.df)=
  c("SOBARCODE.1", "BARCODE.1", "XPOS.1", "YPOS.1", "WELL_USAGE.1",
    "Compound.1", "Intensity.1", "log2Intensity.1",
    "SOBARCODE.2", "BARCODE.2", "XPOS.2", "YPOS.2", "WELL_USAGE.2",
    "Compound.2", "Intensity.2", "log2Intensity.2")
dualFlashlight.fn(data.df, wellName="WELL_USAGE.1", x.name="log2Intensity.1",
                  y.name="log2Intensity.2",
                  sampleName="Sample", sampleColor="black",
                  controls = c("negCTRL", "posCTRL1", "mock1"),
                  controlColors = c("green", "red", "lightblue"),
                  xlab="log2 intensity in plate 1",
                  ylab="log2 intensity in plate 2",
                  main="Plate Pair Correlation Plot", x.legend=NA,
                  y.legend=NA, cex.point=1, cex.legend = 0.8 )
abline(0,1)
```

HTSdata *HTSdata*

Description

Modified raw data from an HTS experiment

Usage

```
data(HTSdata)
```

Format

A data frame with 7116 observations on the following 8 variables.

`SOBARCODE` a factor with levels `SO000001` `SO000002`

`BARCODE` a factor with levels `PL000001` `PL000002` `PL000003` `PL000004` `PL000005` `PL000006` `PL000007` `PL000008` `PL000009` `PL000010` `PL000011` `PL000012`

`XPOS` a numeric vector

`YPOS` a numeric vector

`WELL_USAGE` a factor with levels `mock1` `mock2` `negCTRL` `posCTRL1` `posCTRL2` `posCTRL3` `Sample`

`Compound` a factor with levels `-` `-` `Cmpd100001` `Cmpd100002` `Cmpd100003` `Cmpd100004` `Cmpd100005` `Cmpd100006` `Cmpd100007` `Cmpd100008` `Cmpd100009` `Cmpd100010` `Cmpd100011` `Cmpd100012` `Cmpd100013` `Cmpd100014` `Cmpd100015` `Cmpd100016` `Cmpd100017` `Cmpd100018` `Cmpd100019` `Cmpd100020` `Cmpd100021` `Cmpd100022` `Cmpd100023` `Cmpd100024` `Cmpd100025` `Cmpd100026` `Cmpd100027` `Cmpd100028` `Cmpd100029` `Cmpd100030` `Cmpd100031` `Cmpd100032` `Cmpd100033` `Cmpd100034` `Cmpd100035` `Cmpd100036` `Cmpd100037` `Cmpd100038` `Cmpd100039` `Cmpd100040` `Cmpd100041` `Cmpd100042` `Cmpd100043` `Cmpd100044` `Cmpd100045`

```
Cmpd100046 Cmpd100047 Cmpd100048 Cmpd100049 Cmpd100050 Cmpd100051 Cmpd100052
Cmpd100053 Cmpd100054 Cmpd100055 Cmpd100056 Cmpd100057 Cmpd100058 Cmpd100059
Cmpd100060 Cmpd100061 Cmpd100062 Cmpd100063 Cmpd100064 Cmpd100065 Cmpd100066
Cmpd100067 Cmpd100068 Cmpd100069 Cmpd100070 Cmpd100071 Cmpd100072 Cmpd100073
Cmpd100074 Cmpd100075 Cmpd100076 Cmpd100077 Cmpd100078 Cmpd100079 Cmpd100080
Cmpd100081 Cmpd100082 Cmpd100083 Cmpd100084 Cmpd100085 Cmpd100086 Cmpd100087
Cmpd100088 Cmpd100089 Cmpd100090 Cmpd100091 Cmpd100092 Cmpd100093 Cmpd100094
Cmpd100095 Cmpd100096 Cmpd100097 Cmpd100098 Cmpd100099 Cmpd100100 Cmpd100101
Cmpd100102 Cmpd100103 Cmpd100104 Cmpd100105 Cmpd100106 Cmpd100107 Cmpd100108
Cmpd100109 Cmpd100110 Cmpd100111 Cmpd100112 Cmpd100113 Cmpd100114 Cmpd100115
Cmpd100116 Cmpd100117 Cmpd100118 Cmpd100119 Cmpd100120 Cmpd100121 Cmpd100122
Cmpd100123 Cmpd100124 Cmpd100125 Cmpd100126 Cmpd100127 Cmpd100128 Cmpd100129
Cmpd100130 Cmpd100131 Cmpd100132 Cmpd100133 Cmpd100134 Cmpd100135 Cmpd100136
Cmpd100137 Cmpd100138 Cmpd100139 Cmpd100140 Cmpd100141 Cmpd100142 Cmpd100143
Cmpd100144 Cmpd100145 Cmpd100146 Cmpd100147 Cmpd100148 Cmpd100149 Cmpd100150
Cmpd100151 Cmpd100152 Cmpd100153 Cmpd100154 Cmpd100155 Cmpd100156 Cmpd100157
Cmpd100158 Cmpd100159 Cmpd100160 Cmpd100161 Cmpd100162 Cmpd100163 Cmpd100164
Cmpd100165 Cmpd100166 Cmpd100167 Cmpd100168 Cmpd100169 Cmpd100170 Cmpd100171
Cmpd100172 Cmpd100173 Cmpd100174 Cmpd100175 Cmpd100176 Cmpd100177 Cmpd100178
Cmpd100179 Cmpd100180 Cmpd100181 Cmpd100182 Cmpd100183 Cmpd100184 Cmpd100185
Cmpd100186 Cmpd100187 Cmpd100188 Cmpd100189 Cmpd100190 Cmpd100191 Cmpd100192
Cmpd100193 Cmpd100194 Cmpd100195 Cmpd100196 Cmpd100197 Cmpd100198 Cmpd100199
Cmpd100200 Cmpd100201 Cmpd100202 Cmpd100203 Cmpd100204 Cmpd100205 Cmpd100206
Cmpd100207 Cmpd100208 Cmpd100209 Cmpd100210 Cmpd100211 Cmpd100212 Cmpd100213
Cmpd100214 Cmpd100215 Cmpd100216 Cmpd100217 Cmpd100218 Cmpd100219 Cmpd100220
Cmpd100221 Cmpd100222 Cmpd100223 Cmpd100224 Cmpd100225 Cmpd100226 Cmpd100227
Cmpd100228 Cmpd100229 Cmpd100230 Cmpd100231 Cmpd100232 Cmpd100233 Cmpd100234
Cmpd100235 Cmpd100236 Cmpd100237 Cmpd100238 Cmpd100239 Cmpd100240 Cmpd100241
Cmpd100242 Cmpd100243 Cmpd100244 Cmpd100245 Cmpd100246 Cmpd100247 Cmpd100248
Cmpd100249 Cmpd100250 Cmpd100251 Cmpd100252 Cmpd100253 Cmpd100254 Cmpd100255
Cmpd100256 Cmpd100257 Cmpd100258 Cmpd100259 Cmpd100260 Cmpd100261 Cmpd100262
Cmpd100263 Cmpd100264 Cmpd100265 Cmpd100266 Cmpd100267 Cmpd100268 Cmpd100269
Cmpd100270 Cmpd100271 Cmpd100272 Cmpd100273 Cmpd100274 Cmpd100275 Cmpd100276
Cmpd100277 Cmpd100278 Cmpd100279 Cmpd100280 Cmpd100281 Cmpd100282 Cmpd100283
Cmpd100284 Cmpd100285 Cmpd100286 Cmpd100287 Cmpd100288 Cmpd100289 Cmpd100290
Cmpd100291 Cmpd100292 Cmpd100293 Cmpd100294 Cmpd100295 Cmpd100296 Cmpd100297
Cmpd100298 Cmpd100299 Cmpd100300 Cmpd100301 Cmpd100302 Cmpd100303 Cmpd100304
Cmpd100305 Cmpd100306 Cmpd100307 Cmpd100308 Cmpd100309 Cmpd100310 Cmpd100311
Cmpd100312 Cmpd100313 Cmpd100314 Cmpd100315 Cmpd100316 Cmpd100317 Cmpd100318
Cmpd100319 Cmpd100320 Cmpd100321 Cmpd100322 Cmpd100323 Cmpd100324 Cmpd100325
Cmpd100326 Cmpd100327 Cmpd100328 Cmpd100329 Cmpd100330 Cmpd100331 Cmpd100332
Cmpd100333 Cmpd100334 Cmpd100335 Cmpd100336 Cmpd100337 Cmpd100338 Cmpd100339
Cmpd100340 Cmpd100341 Cmpd100342 Cmpd100343 Cmpd100344 Cmpd100345 Cmpd100346
Cmpd100347 Cmpd100348 Cmpd100349 Cmpd100350 Cmpd100351 Cmpd100352 Cmpd100353
Cmpd100354 Cmpd100355 Cmpd100356 Cmpd100357 Cmpd100358 Cmpd100359 Cmpd100360
Cmpd100361 Cmpd100362 Cmpd100363 Cmpd100364 Cmpd100365 Cmpd100366 Cmpd100367
Cmpd100368 Cmpd100369 Cmpd100370 Cmpd100371 Cmpd100372 Cmpd100373 Cmpd100374
Cmpd100375 Cmpd100376 Cmpd100377 Cmpd100378 Cmpd100379 Cmpd100380 Cmpd100381
```

Cmpd100382 Cmpd100383 Cmpd100384 Cmpd100385 Cmpd100386 Cmpd100387 Cmpd100388
Cmpd100389 Cmpd100390 Cmpd100391 Cmpd100392 Cmpd100393 Cmpd100394 Cmpd100395
Cmpd100396 Cmpd100397 Cmpd100398 Cmpd100399 Cmpd100400 Cmpd100401 Cmpd100402
Cmpd100403 Cmpd100404 Cmpd100405 Cmpd100406 Cmpd100407 Cmpd100408 Cmpd100409
Cmpd100410 Cmpd100411 Cmpd100412 Cmpd100413 Cmpd100414 Cmpd100415 Cmpd100416
Cmpd100417 Cmpd100418 Cmpd100419 Cmpd100420 Cmpd100421 Cmpd100422 Cmpd100423
Cmpd100424 Cmpd100425 Cmpd100426 Cmpd100427 Cmpd100428 Cmpd100429 Cmpd100430
Cmpd100431 Cmpd100432 Cmpd100433 Cmpd100434 Cmpd100435 Cmpd100436 Cmpd100437
Cmpd100438 Cmpd100439 Cmpd100440 Cmpd100441 Cmpd100442 Cmpd100443 Cmpd100444
Cmpd100445 Cmpd100446 Cmpd100447 Cmpd100448 Cmpd100449 Cmpd100450 Cmpd100451
Cmpd100452 Cmpd100453 Cmpd100454 Cmpd100455 Cmpd100456 Cmpd100457 Cmpd100458
Cmpd100459 Cmpd100460 Cmpd100461 Cmpd100462 Cmpd100463 Cmpd100464 Cmpd100465
Cmpd100466 Cmpd100467 Cmpd100468 Cmpd100469 Cmpd100470 Cmpd100471 Cmpd100472
Cmpd100473 Cmpd100474 Cmpd100475 Cmpd100476 Cmpd100477 Cmpd100478 Cmpd100479
Cmpd100480 Cmpd100481 Cmpd100482 Cmpd100483 Cmpd100484 Cmpd100485 Cmpd100486
Cmpd100487 Cmpd100488 Cmpd100489 Cmpd100490 Cmpd100491 Cmpd100492 Cmpd100493
Cmpd100494 Cmpd100495 Cmpd100496 Cmpd100497 Cmpd100498 Cmpd100499 Cmpd100500
Cmpd100501 Cmpd100502 Cmpd100503 Cmpd100504 Cmpd100505 Cmpd100506 Cmpd100507
Cmpd100508 Cmpd100509 Cmpd100510 Cmpd100511 Cmpd100512 Cmpd100513 Cmpd100514
Cmpd100515 Cmpd100516 Cmpd100517 Cmpd100518 Cmpd100519 Cmpd100520 Cmpd100521
Cmpd100522 Cmpd100523 Cmpd100524 Cmpd100525 Cmpd100526 Cmpd100527 Cmpd100528
Cmpd100529 Cmpd100530 Cmpd100531 Cmpd100532 Cmpd100533 Cmpd100534 Cmpd100535
Cmpd100536 Cmpd100537 Cmpd100538 Cmpd100539 Cmpd100540 Cmpd100541 Cmpd100542
Cmpd100543 Cmpd100544 Cmpd100545 Cmpd100546 Cmpd100547 Cmpd100548 Cmpd100549
Cmpd100550 Cmpd100551 Cmpd100552 Cmpd100553 Cmpd100554 Cmpd100555 Cmpd100556
Cmpd100557 Cmpd100558 Cmpd100559 Cmpd100560 Cmpd100561 Cmpd100562 Cmpd100563
Cmpd100564 Cmpd100565 Cmpd100566 Cmpd100567 Cmpd100568 Cmpd100569 Cmpd100570
Cmpd100571 Cmpd100572 Cmpd100573 Cmpd100574 Cmpd100575 Cmpd100576 Cmpd100577
Cmpd100578 Cmpd100579 Cmpd100580 Cmpd100581 Cmpd100582 Cmpd100583 Cmpd100584
Cmpd100585 Cmpd100586 Cmpd100587 Cmpd100588 Cmpd100589 Cmpd100590 Cmpd100591
Cmpd100592 Cmpd100593 Cmpd100594 Cmpd100595 Cmpd100596 Cmpd100597 Cmpd100598
Cmpd100599 Cmpd100600 Cmpd100601 Cmpd100602 Cmpd100603 Cmpd100604 Cmpd100605
Cmpd100606 Cmpd100607 Cmpd100608 Cmpd100609 Cmpd100610 Cmpd100611 Cmpd100612
Cmpd100613 Cmpd100614 Cmpd100615 Cmpd100616 Cmpd100617 Cmpd100618 Cmpd100619
Cmpd100620 Cmpd100621 Cmpd100622 Cmpd100623 Cmpd100624 Cmpd100625 Cmpd100626
Cmpd100627 Cmpd100628 Cmpd100629 Cmpd100630 Cmpd100631 Cmpd100632

`Intensity` a numeric vector

`log2Intensity` a numeric vector

Details

Modified raw data from an HTS experiment for demonstrating the utility of our package displayHTS

Examples

```
data(HTSdata)
boxplot(HTSdata[,"log2Intensity"] ~ HTSdata[, "WELL_USAGE"], cex.axis=0.75)
```

HTSdataSort *An HTS dataset sorted from HTSdata*

Description

Data sorted by source plate, plate, row and column numbers

Usage

```
data(HTSdataSort)
```

Format

A data frame with 4608 observations on the following 8 variables.

SOBARCODE a factor with levels SO000001 SO000002

BARCODE a factor with levels PL000001 PL000002 PL000003 PL000004 PL000005 PL000006 PL000007 PL000008 PL000009 PL000010 PL000011 PL000012

XPOS a numeric vector

YPOS a numeric vector

WELL_USAGE a factor with levels mock1 mock2 negCTRL posCTRL1 posCTRL2 posCTRL3 Sample

Compound a factor with levels - - Cmpd100001 Cmpd100002 Cmpd100003 Cmpd100004 Cmpd100005 Cmpd100006 Cmpd100007 Cmpd100008 Cmpd100009 Cmpd100010 Cmpd100011 Cmpd100012 Cmpd100013 Cmpd100014 Cmpd100015 Cmpd100016 Cmpd100017 Cmpd100018 Cmpd100019 Cmpd100020 Cmpd100021 Cmpd100022 Cmpd100023 Cmpd100024 Cmpd100025 Cmpd100026 Cmpd100027 Cmpd100028 Cmpd100029 Cmpd100030 Cmpd100031 Cmpd100032 Cmpd100033 Cmpd100034 Cmpd100035 Cmpd100036 Cmpd100037 Cmpd100038 Cmpd100039 Cmpd100040 Cmpd100041 Cmpd100042 Cmpd100043 Cmpd100044 Cmpd100045 Cmpd100046 Cmpd100047 Cmpd100048 Cmpd100049 Cmpd100050 Cmpd100051 Cmpd100052 Cmpd100053 Cmpd100054 Cmpd100055 Cmpd100056 Cmpd100057 Cmpd100058 Cmpd100059 Cmpd100060 Cmpd100061 Cmpd100062 Cmpd100063 Cmpd100064 Cmpd100065 Cmpd100066 Cmpd100067 Cmpd100068 Cmpd100069 Cmpd100070 Cmpd100071 Cmpd100072 Cmpd100073 Cmpd100074 Cmpd100075 Cmpd100076 Cmpd100077 Cmpd100078 Cmpd100079 Cmpd100080 Cmpd100081 Cmpd100082 Cmpd100083 Cmpd100084 Cmpd100085 Cmpd100086 Cmpd100087 Cmpd100088 Cmpd100089 Cmpd100090 Cmpd100091 Cmpd100092 Cmpd100093 Cmpd100094 Cmpd100095 Cmpd100096 Cmpd100097 Cmpd100098 Cmpd100099 Cmpd100100 Cmpd100101 Cmpd100102 Cmpd100103 Cmpd100104 Cmpd100105 Cmpd100106 Cmpd100107 Cmpd100108 Cmpd100109 Cmpd100110 Cmpd100111 Cmpd100112 Cmpd100113 Cmpd100114 Cmpd100115 Cmpd100116 Cmpd100117 Cmpd100118 Cmpd100119 Cmpd100120 Cmpd100121 Cmpd100122 Cmpd100123 Cmpd100124 Cmpd100125 Cmpd100126 Cmpd100127 Cmpd100128 Cmpd100129 Cmpd100130 Cmpd100131 Cmpd100132 Cmpd100133 Cmpd100134 Cmpd100135 Cmpd100136 Cmpd100137 Cmpd100138 Cmpd100139 Cmpd100140 Cmpd100141 Cmpd100142 Cmpd100143 Cmpd100144 Cmpd100145 Cmpd100146 Cmpd100147 Cmpd100148 Cmpd100149 Cmpd100150 Cmpd100151 Cmpd100152 Cmpd100153 Cmpd100154 Cmpd100155 Cmpd100156 Cmpd100157 Cmpd100158 Cmpd100159 Cmpd100160 Cmpd100161 Cmpd100162 Cmpd100163 Cmpd100164 Cmpd100165 Cmpd100166 Cmpd100167 Cmpd100168 Cmpd100169 Cmpd100170 Cmpd100171

```
Cmpd100172 Cmpd100173 Cmpd100174 Cmpd100175 Cmpd100176 Cmpd100177 Cmpd100178
Cmpd100179 Cmpd100180 Cmpd100181 Cmpd100182 Cmpd100183 Cmpd100184 Cmpd100185
Cmpd100186 Cmpd100187 Cmpd100188 Cmpd100189 Cmpd100190 Cmpd100191 Cmpd100192
Cmpd100193 Cmpd100194 Cmpd100195 Cmpd100196 Cmpd100197 Cmpd100198 Cmpd100199
Cmpd100200 Cmpd100201 Cmpd100202 Cmpd100203 Cmpd100204 Cmpd100205 Cmpd100206
Cmpd100207 Cmpd100208 Cmpd100209 Cmpd100210 Cmpd100211 Cmpd100212 Cmpd100213
Cmpd100214 Cmpd100215 Cmpd100216 Cmpd100217 Cmpd100218 Cmpd100219 Cmpd100220
Cmpd100221 Cmpd100222 Cmpd100223 Cmpd100224 Cmpd100225 Cmpd100226 Cmpd100227
Cmpd100228 Cmpd100229 Cmpd100230 Cmpd100231 Cmpd100232 Cmpd100233 Cmpd100234
Cmpd100235 Cmpd100236 Cmpd100237 Cmpd100238 Cmpd100239 Cmpd100240 Cmpd100241
Cmpd100242 Cmpd100243 Cmpd100244 Cmpd100245 Cmpd100246 Cmpd100247 Cmpd100248
Cmpd100249 Cmpd100250 Cmpd100251 Cmpd100252 Cmpd100253 Cmpd100254 Cmpd100255
Cmpd100256 Cmpd100257 Cmpd100258 Cmpd100259 Cmpd100260 Cmpd100261 Cmpd100262
Cmpd100263 Cmpd100264 Cmpd100265 Cmpd100266 Cmpd100267 Cmpd100268 Cmpd100269
Cmpd100270 Cmpd100271 Cmpd100272 Cmpd100273 Cmpd100274 Cmpd100275 Cmpd100276
Cmpd100277 Cmpd100278 Cmpd100279 Cmpd100280 Cmpd100281 Cmpd100282 Cmpd100283
Cmpd100284 Cmpd100285 Cmpd100286 Cmpd100287 Cmpd100288 Cmpd100289 Cmpd100290
Cmpd100291 Cmpd100292 Cmpd100293 Cmpd100294 Cmpd100295 Cmpd100296 Cmpd100297
Cmpd100298 Cmpd100299 Cmpd100300 Cmpd100301 Cmpd100302 Cmpd100303 Cmpd100304
Cmpd100305 Cmpd100306 Cmpd100307 Cmpd100308 Cmpd100309 Cmpd100310 Cmpd100311
Cmpd100312 Cmpd100313 Cmpd100314 Cmpd100315 Cmpd100316 Cmpd100317 Cmpd100318
Cmpd100319 Cmpd100320 Cmpd100321 Cmpd100322 Cmpd100323 Cmpd100324 Cmpd100325
Cmpd100326 Cmpd100327 Cmpd100328 Cmpd100329 Cmpd100330 Cmpd100331 Cmpd100332
Cmpd100333 Cmpd100334 Cmpd100335 Cmpd100336 Cmpd100337 Cmpd100338 Cmpd100339
Cmpd100340 Cmpd100341 Cmpd100342 Cmpd100343 Cmpd100344 Cmpd100345 Cmpd100346
Cmpd100347 Cmpd100348 Cmpd100349 Cmpd100350 Cmpd100351 Cmpd100352 Cmpd100353
Cmpd100354 Cmpd100355 Cmpd100356 Cmpd100357 Cmpd100358 Cmpd100359 Cmpd100360
Cmpd100361 Cmpd100362 Cmpd100363 Cmpd100364 Cmpd100365 Cmpd100366 Cmpd100367
Cmpd100368 Cmpd100369 Cmpd100370 Cmpd100371 Cmpd100372 Cmpd100373 Cmpd100374
Cmpd100375 Cmpd100376 Cmpd100377 Cmpd100378 Cmpd100379 Cmpd100380 Cmpd100381
Cmpd100382 Cmpd100383 Cmpd100384 Cmpd100385 Cmpd100386 Cmpd100387 Cmpd100388
Cmpd100389 Cmpd100390 Cmpd100391 Cmpd100392 Cmpd100393 Cmpd100394 Cmpd100395
Cmpd100396 Cmpd100397 Cmpd100398 Cmpd100399 Cmpd100400 Cmpd100401 Cmpd100402
Cmpd100403 Cmpd100404 Cmpd100405 Cmpd100406 Cmpd100407 Cmpd100408 Cmpd100409
Cmpd100410 Cmpd100411 Cmpd100412 Cmpd100413 Cmpd100414 Cmpd100415 Cmpd100416
Cmpd100417 Cmpd100418 Cmpd100419 Cmpd100420 Cmpd100421 Cmpd100422 Cmpd100423
Cmpd100424 Cmpd100425 Cmpd100426 Cmpd100427 Cmpd100428 Cmpd100429 Cmpd100430
Cmpd100431 Cmpd100432 Cmpd100433 Cmpd100434 Cmpd100435 Cmpd100436 Cmpd100437
Cmpd100438 Cmpd100439 Cmpd100440 Cmpd100441 Cmpd100442 Cmpd100443 Cmpd100444
Cmpd100445 Cmpd100446 Cmpd100447 Cmpd100448 Cmpd100449 Cmpd100450 Cmpd100451
Cmpd100452 Cmpd100453 Cmpd100454 Cmpd100455 Cmpd100456 Cmpd100457 Cmpd100458
Cmpd100459 Cmpd100460 Cmpd100461 Cmpd100462 Cmpd100463 Cmpd100464 Cmpd100465
Cmpd100466 Cmpd100467 Cmpd100468 Cmpd100469 Cmpd100470 Cmpd100471 Cmpd100472
Cmpd100473 Cmpd100474 Cmpd100475 Cmpd100476 Cmpd100477 Cmpd100478 Cmpd100479
Cmpd100480 Cmpd100481 Cmpd100482 Cmpd100483 Cmpd100484 Cmpd100485 Cmpd100486
Cmpd100487 Cmpd100488 Cmpd100489 Cmpd100490 Cmpd100491 Cmpd100492 Cmpd100493
Cmpd100494 Cmpd100495 Cmpd100496 Cmpd100497 Cmpd100498 Cmpd100499 Cmpd100500
Cmpd100501 Cmpd100502 Cmpd100503 Cmpd100504 Cmpd100505 Cmpd100506 Cmpd100507
```

Cmpd100508 Cmpd100509 Cmpd100510 Cmpd100511 Cmpd100512 Cmpd100513 Cmpd100514
Cmpd100515 Cmpd100516 Cmpd100517 Cmpd100518 Cmpd100519 Cmpd100520 Cmpd100521
Cmpd100522 Cmpd100523 Cmpd100524 Cmpd100525 Cmpd100526 Cmpd100527 Cmpd100528
Cmpd100529 Cmpd100530 Cmpd100531 Cmpd100532 Cmpd100533 Cmpd100534 Cmpd100535
Cmpd100536 Cmpd100537 Cmpd100538 Cmpd100539 Cmpd100540 Cmpd100541 Cmpd100542
Cmpd100543 Cmpd100544 Cmpd100545 Cmpd100546 Cmpd100547 Cmpd100548 Cmpd100549
Cmpd100550 Cmpd100551 Cmpd100552 Cmpd100553 Cmpd100554 Cmpd100555 Cmpd100556
Cmpd100557 Cmpd100558 Cmpd100559 Cmpd100560 Cmpd100561 Cmpd100562 Cmpd100563
Cmpd100564 Cmpd100565 Cmpd100566 Cmpd100567 Cmpd100568 Cmpd100569 Cmpd100570
Cmpd100571 Cmpd100572 Cmpd100573 Cmpd100574 Cmpd100575 Cmpd100576 Cmpd100577
Cmpd100578 Cmpd100579 Cmpd100580 Cmpd100581 Cmpd100582 Cmpd100583 Cmpd100584
Cmpd100585 Cmpd100586 Cmpd100587 Cmpd100588 Cmpd100589 Cmpd100590 Cmpd100591
Cmpd100592 Cmpd100593 Cmpd100594 Cmpd100595 Cmpd100596 Cmpd100597 Cmpd100598
Cmpd100599 Cmpd100600 Cmpd100601 Cmpd100602 Cmpd100603 Cmpd100604 Cmpd100605
Cmpd100606 Cmpd100607 Cmpd100608 Cmpd100609 Cmpd100610 Cmpd100611 Cmpd100612
Cmpd100613 Cmpd100614 Cmpd100615 Cmpd100616 Cmpd100617 Cmpd100618 Cmpd100619
Cmpd100620 Cmpd100621 Cmpd100622 Cmpd100623 Cmpd100624 Cmpd100625 Cmpd100626
Cmpd100627 Cmpd100628 Cmpd100629 Cmpd100630 Cmpd100631 Cmpd100632

`Intensity` a numeric vector

`log2Intensity` a numeric vector

Details

Sorted data from an HTS experiment for demonstrating the utility of package displayHTS

Examples

```
data(HTSdataSort)
boxplot(HTSdataSort[,"log2Intensity"] ~ HTSdataSort[, "WELL_USAGE"],
        cex.axis=0.75)
```

`HTSresults` *Calculated results from an HTS experiments*

Description

Calculated results including SSMD, p-value etc from an HTS experiment

Usage

```
data(HTSresults)
```

Format

A data frame with 768 observations on the following 9 variables.

`SOBARCODE` a factor with levels `SO000001` `SO000002`

`XPOS` a numeric vector

`YPOS` a numeric vector

`WELL_USAGE` a factor with levels `mock1` `mock2` `negCTRL` `posCTRL1` `posCTRL2` `posCTRL3` `Sample`

`Compound` a factor with levels - - Cmpd100001 Cmpd100002 Cmpd100003
Cmpd100004 Cmpd100005 Cmpd100006 Cmpd100007 Cmpd100008 Cmpd100009 Cmpd100010
Cmpd100011 Cmpd100012 Cmpd100013 Cmpd100014 Cmpd100015 Cmpd100016 Cmpd100017
Cmpd100018 Cmpd100019 Cmpd100020 Cmpd100021 Cmpd100022 Cmpd100023 Cmpd100024
Cmpd100025 Cmpd100026 Cmpd100027 Cmpd100028 Cmpd100029 Cmpd100030 Cmpd100031
Cmpd100032 Cmpd100033 Cmpd100034 Cmpd100035 Cmpd100036 Cmpd100037 Cmpd100038
Cmpd100039 Cmpd100040 Cmpd100041 Cmpd100042 Cmpd100043 Cmpd100044 Cmpd100045
Cmpd100046 Cmpd100047 Cmpd100048 Cmpd100049 Cmpd100050 Cmpd100051 Cmpd100052
Cmpd100053 Cmpd100054 Cmpd100055 Cmpd100056 Cmpd100057 Cmpd100058 Cmpd100059
Cmpd100060 Cmpd100061 Cmpd100062 Cmpd100063 Cmpd100064 Cmpd100065 Cmpd100066
Cmpd100067 Cmpd100068 Cmpd100069 Cmpd100070 Cmpd100071 Cmpd100072 Cmpd100073
Cmpd100074 Cmpd100075 Cmpd100076 Cmpd100077 Cmpd100078 Cmpd100079 Cmpd100080
Cmpd100081 Cmpd100082 Cmpd100083 Cmpd100084 Cmpd100085 Cmpd100086 Cmpd100087
Cmpd100088 Cmpd100089 Cmpd100090 Cmpd100091 Cmpd100092 Cmpd100093 Cmpd100094
Cmpd100095 Cmpd100096 Cmpd100097 Cmpd100098 Cmpd100099 Cmpd100100 Cmpd100101
Cmpd100102 Cmpd100103 Cmpd100104 Cmpd100105 Cmpd100106 Cmpd100107 Cmpd100108
Cmpd100109 Cmpd100110 Cmpd100111 Cmpd100112 Cmpd100113 Cmpd100114 Cmpd100115
Cmpd100116 Cmpd100117 Cmpd100118 Cmpd100119 Cmpd100120 Cmpd100121 Cmpd100122
Cmpd100123 Cmpd100124 Cmpd100125 Cmpd100126 Cmpd100127 Cmpd100128 Cmpd100129
Cmpd100130 Cmpd100131 Cmpd100132 Cmpd100133 Cmpd100134 Cmpd100135 Cmpd100136
Cmpd100137 Cmpd100138 Cmpd100139 Cmpd100140 Cmpd100141 Cmpd100142 Cmpd100143
Cmpd100144 Cmpd100145 Cmpd100146 Cmpd100147 Cmpd100148 Cmpd100149 Cmpd100150
Cmpd100151 Cmpd100152 Cmpd100153 Cmpd100154 Cmpd100155 Cmpd100156 Cmpd100157
Cmpd100158 Cmpd100159 Cmpd100160 Cmpd100161 Cmpd100162 Cmpd100163 Cmpd100164
Cmpd100165 Cmpd100166 Cmpd100167 Cmpd100168 Cmpd100169 Cmpd100170 Cmpd100171
Cmpd100172 Cmpd100173 Cmpd100174 Cmpd100175 Cmpd100176 Cmpd100177 Cmpd100178
Cmpd100179 Cmpd100180 Cmpd100181 Cmpd100182 Cmpd100183 Cmpd100184 Cmpd100185
Cmpd100186 Cmpd100187 Cmpd100188 Cmpd100189 Cmpd100190 Cmpd100191 Cmpd100192
Cmpd100193 Cmpd100194 Cmpd100195 Cmpd100196 Cmpd100197 Cmpd100198 Cmpd100199
Cmpd100200 Cmpd100201 Cmpd100202 Cmpd100203 Cmpd100204 Cmpd100205 Cmpd100206
Cmpd100207 Cmpd100208 Cmpd100209 Cmpd100210 Cmpd100211 Cmpd100212 Cmpd100213
Cmpd100214 Cmpd100215 Cmpd100216 Cmpd100217 Cmpd100218 Cmpd100219 Cmpd100220
Cmpd100221 Cmpd100222 Cmpd100223 Cmpd100224 Cmpd100225 Cmpd100226 Cmpd100227
Cmpd100228 Cmpd100229 Cmpd100230 Cmpd100231 Cmpd100232 Cmpd100233 Cmpd100234
Cmpd100235 Cmpd100236 Cmpd100237 Cmpd100238 Cmpd100239 Cmpd100240 Cmpd100241
Cmpd100242 Cmpd100243 Cmpd100244 Cmpd100245 Cmpd100246 Cmpd100247 Cmpd100248
Cmpd100249 Cmpd100250 Cmpd100251 Cmpd100252 Cmpd100253 Cmpd100254 Cmpd100255
Cmpd100256 Cmpd100257 Cmpd100258 Cmpd100259 Cmpd100260 Cmpd100261 Cmpd100262
Cmpd100263 Cmpd100264 Cmpd100265 Cmpd100266 Cmpd100267 Cmpd100268 Cmpd100269
Cmpd100270 Cmpd100271 Cmpd100272 Cmpd100273 Cmpd100274 Cmpd100275 Cmpd100276

```
Cmpd100277 Cmpd100278 Cmpd100279 Cmpd100280 Cmpd100281 Cmpd100282 Cmpd100283
Cmpd100284 Cmpd100285 Cmpd100286 Cmpd100287 Cmpd100288 Cmpd100289 Cmpd100290
Cmpd100291 Cmpd100292 Cmpd100293 Cmpd100294 Cmpd100295 Cmpd100296 Cmpd100297
Cmpd100298 Cmpd100299 Cmpd100300 Cmpd100301 Cmpd100302 Cmpd100303 Cmpd100304
Cmpd100305 Cmpd100306 Cmpd100307 Cmpd100308 Cmpd100309 Cmpd100310 Cmpd100311
Cmpd100312 Cmpd100313 Cmpd100314 Cmpd100315 Cmpd100316 Cmpd100317 Cmpd100318
Cmpd100319 Cmpd100320 Cmpd100321 Cmpd100322 Cmpd100323 Cmpd100324 Cmpd100325
Cmpd100326 Cmpd100327 Cmpd100328 Cmpd100329 Cmpd100330 Cmpd100331 Cmpd100332
Cmpd100333 Cmpd100334 Cmpd100335 Cmpd100336 Cmpd100337 Cmpd100338 Cmpd100339
Cmpd100340 Cmpd100341 Cmpd100342 Cmpd100343 Cmpd100344 Cmpd100345 Cmpd100346
Cmpd100347 Cmpd100348 Cmpd100349 Cmpd100350 Cmpd100351 Cmpd100352 Cmpd100353
Cmpd100354 Cmpd100355 Cmpd100356 Cmpd100357 Cmpd100358 Cmpd100359 Cmpd100360
Cmpd100361 Cmpd100362 Cmpd100363 Cmpd100364 Cmpd100365 Cmpd100366 Cmpd100367
Cmpd100368 Cmpd100369 Cmpd100370 Cmpd100371 Cmpd100372 Cmpd100373 Cmpd100374
Cmpd100375 Cmpd100376 Cmpd100377 Cmpd100378 Cmpd100379 Cmpd100380 Cmpd100381
Cmpd100382 Cmpd100383 Cmpd100384 Cmpd100385 Cmpd100386 Cmpd100387 Cmpd100388
Cmpd100389 Cmpd100390 Cmpd100391 Cmpd100392 Cmpd100393 Cmpd100394 Cmpd100395
Cmpd100396 Cmpd100397 Cmpd100398 Cmpd100399 Cmpd100400 Cmpd100401 Cmpd100402
Cmpd100403 Cmpd100404 Cmpd100405 Cmpd100406 Cmpd100407 Cmpd100408 Cmpd100409
Cmpd100410 Cmpd100411 Cmpd100412 Cmpd100413 Cmpd100414 Cmpd100415 Cmpd100416
Cmpd100417 Cmpd100418 Cmpd100419 Cmpd100420 Cmpd100421 Cmpd100422 Cmpd100423
Cmpd100424 Cmpd100425 Cmpd100426 Cmpd100427 Cmpd100428 Cmpd100429 Cmpd100430
Cmpd100431 Cmpd100432 Cmpd100433 Cmpd100434 Cmpd100435 Cmpd100436 Cmpd100437
Cmpd100438 Cmpd100439 Cmpd100440 Cmpd100441 Cmpd100442 Cmpd100443 Cmpd100444
Cmpd100445 Cmpd100446 Cmpd100447 Cmpd100448 Cmpd100449 Cmpd100450 Cmpd100451
Cmpd100452 Cmpd100453 Cmpd100454 Cmpd100455 Cmpd100456 Cmpd100457 Cmpd100458
Cmpd100459 Cmpd100460 Cmpd100461 Cmpd100462 Cmpd100463 Cmpd100464 Cmpd100465
Cmpd100466 Cmpd100467 Cmpd100468 Cmpd100469 Cmpd100470 Cmpd100471 Cmpd100472
Cmpd100473 Cmpd100474 Cmpd100475 Cmpd100476 Cmpd100477 Cmpd100478 Cmpd100479
Cmpd100480 Cmpd100481 Cmpd100482 Cmpd100483 Cmpd100484 Cmpd100485 Cmpd100486
Cmpd100487 Cmpd100488 Cmpd100489 Cmpd100490 Cmpd100491 Cmpd100492 Cmpd100493
Cmpd100494 Cmpd100495 Cmpd100496 Cmpd100497 Cmpd100498 Cmpd100499 Cmpd100500
Cmpd100501 Cmpd100502 Cmpd100503 Cmpd100504 Cmpd100505 Cmpd100506 Cmpd100507
Cmpd100508 Cmpd100509 Cmpd100510 Cmpd100511 Cmpd100512 Cmpd100513 Cmpd100514
Cmpd100515 Cmpd100516 Cmpd100517 Cmpd100518 Cmpd100519 Cmpd100520 Cmpd100521
Cmpd100522 Cmpd100523 Cmpd100524 Cmpd100525 Cmpd100526 Cmpd100527 Cmpd100528
Cmpd100529 Cmpd100530 Cmpd100531 Cmpd100532 Cmpd100533 Cmpd100534 Cmpd100535
Cmpd100536 Cmpd100537 Cmpd100538 Cmpd100539 Cmpd100540 Cmpd100541 Cmpd100542
Cmpd100543 Cmpd100544 Cmpd100545 Cmpd100546 Cmpd100547 Cmpd100548 Cmpd100549
Cmpd100550 Cmpd100551 Cmpd100552 Cmpd100553 Cmpd100554 Cmpd100555 Cmpd100556
Cmpd100557 Cmpd100558 Cmpd100559 Cmpd100560 Cmpd100561 Cmpd100562 Cmpd100563
Cmpd100564 Cmpd100565 Cmpd100566 Cmpd100567 Cmpd100568 Cmpd100569 Cmpd100570
Cmpd100571 Cmpd100572 Cmpd100573 Cmpd100574 Cmpd100575 Cmpd100576 Cmpd100577
Cmpd100578 Cmpd100579 Cmpd100580 Cmpd100581 Cmpd100582 Cmpd100583 Cmpd100584
Cmpd100585 Cmpd100586 Cmpd100587 Cmpd100588 Cmpd100589 Cmpd100590 Cmpd100591
Cmpd100592 Cmpd100593 Cmpd100594 Cmpd100595 Cmpd100596 Cmpd100597 Cmpd100598
Cmpd100599 Cmpd100600 Cmpd100601 Cmpd100602 Cmpd100603 Cmpd100604 Cmpd100605
Cmpd100606 Cmpd100607 Cmpd100608 Cmpd100609 Cmpd100610 Cmpd100611 Cmpd100612
```

Cmpd100613 Cmpd100614 Cmpd100615 Cmpd100616 Cmpd100617 Cmpd100618 Cmpd100619
Cmpd100620 Cmpd100621 Cmpd100622 Cmpd100623 Cmpd100624 Cmpd100625 Cmpd100626
Cmpd100627 Cmpd100628 Cmpd100629 Cmpd100630 Cmpd100631 Cmpd100632

`ssmd` a numeric vector

`mean` a numeric vector

`p.value` a numeric vector

`Rep` a numeric vector

Details

Results including SSMD, p-value, mean value, replicates from an HTS experiment

Examples

```
data(HTSresults)
plot(HTSresults[,"ssmd"], log10(HTSresults[,"p.value"]))
```

`imageDesign.fn` *Plate Design Image*

Description

A function that displays the image of well designs in a plate.

Usage

```
imageDesign.fn(dataOnePlate.df, wellName = NA, rowName, colName, wells = NULL, colors = NULL, title =
```

Arguments

`dataOnePlate.df`	The data for one plate including at least three columns for well names, rows and columns, respectively.
`wellName`	Name of the column in dataOnePlate.df indicating well types.
`rowName`	Name of the column in dataOnePlate.df indicating row numbers in a plate.
`colName`	Name of the column in dataOnePlate.df indicating column numbers in a plate.
`wells`	Names for unique wells.
`colors`	Colors for corresponding unique wells.
`title`	Title of the image.

Author(s)

Xiaohua Douglas Zhang & Zhaozhi Zhang

References

Zhang XHD, Zhang ZZ. 2013. displayHTS: a R package displaying data and results from high-throughput screening experiments. Bioinformatics (submitted).

Zhang XHD, 2011. Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-scale RNAi Research. Cambridge University Press, Cambridge, UK

Zhang XHD, 2008. Novel analytic criteria and effective plate designs for quality control in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 363-377

Zhang XHD, Espeseth AS, Johnson E, Chin J, Gates A, Mitnaul L, Marine SD, Tian J, Stec EM, Kunapuli P, Holder DJ, Heyse JF, Stulovici B, Ferrer M. 2008. Integrating experimental and analytic approaches to improve data quality in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 378-389

See Also

imageIntensity.fn, dualFlashlight.fn, plateWellSeries.fn

Examples

```
# for control image
 data(HTSdataSort)
 wells = as.character(unique(HTSdataSort[, "WELL_USAGE"])); wells
 colors = c("black",  "yellow", "grey", "blue", "skyblue", "green", "red")
 plate.vec = as.vector(HTSdataSort[,"BARCODE"]); plates=unique(plate.vec)
 data.df = HTSdataSort[plate.vec==plates[1], c("XPOS","YPOS","WELL_USAGE")]
 imageDesign.fn(dataOnePlate.df=data.df, wellName="WELL_USAGE", rowName="XPOS",
                colName="YPOS", wells=wells, colors=colors)

# for hit and control image
 data(HTSresults)
 condtSample = HTSresults[, "WELL_USAGE"] == "Sample"
 condtUp = HTSresults[,"ssmd"] >= 1 & HTSresults[,"mean"] >= log2(1.2)
 condtDown = HTSresults[,"ssmd"] <= -1 & HTSresults[,"mean"] <= -log2(1.2)
 sum(condtSample & (condtUp | condtDown) )/sum(condtSample)
 hit.vec = as.character(HTSresults[, "WELL_USAGE"])
 hit.vec[ condtSample & condtUp ] = "up-hit"
 hit.vec[ condtSample & condtDown ] = "down-hit"
 hit.vec[ condtSample & !condtUp & !condtDown] = "non-hit"
 result.df = cbind(HTSresults, "hitResult"=hit.vec)
 wells = as.character(unique(result.df[, "hitResult"])); wells
 colors = c("black",  "green", "white", "grey", "red",
            "purple1", "purple2", "yellow", "purple3")
 par( mfrow=c(2,1) )
 imageDesign.fn(dataOnePlate.df=result.df[1:384,], wellName="hitResult",
                rowName="XPOS", colName="YPOS", wells=wells, colors=colors,
                title="Source Plate I")
 imageDesign.fn(dataOnePlate.df=result.df[1:384+384,],wellName="hitResult",
               rowName="XPOS", colName="YPOS", wells=wells, colors=colors,
                title="Source Plate II")
```

imageIntensity.fn *Plate Intensity Image*

Description

A function that displays the image of intensity of each well in a plate.

Usage

```
imageIntensity.fn(data.df, intensityName = NA, plateName = "BARCODE", wellName = "WELL_USAGE", rowNam
```

Arguments

data.df	The data for all plates including at least five columns for intensity, plate names, well names, rows and columns, respectively.
intensityName	Name of the column in data.df indicating intensities in a plate.
plateName	Name of the column in data.df indicating plate names.
wellName	Name of the column in data.df indicating well types.
rowName	Name of the column in data.df indicating row numbers in a plate.
colName	Name of the column in data.df indicating column numbers in a plate.
sampleName	Name in well types indicating sample names.
sourcePlateName	Name of the column in data.df indicating soure plate names.

Author(s)

Xiaohua Douglas Zhang & Zhaozhi Zhang

References

Zhang XHD, Zhang ZZ. 2013. displayHTS: a R package displaying data and results from high-throughput screening experiments. Bioinformatics (submitted).

Zhang XHD, 2011. Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-scale RNAi Research. Cambridge University Press, Cambridge, UK.

Zhang XHD, 2008. Novel analytic criteria and effective plate designs for quality control in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 363-377.

Zhang XHD, Espeseth AS, Johnson E, Chin J, Gates A, Mitnaul L, Marine SD, Tian J, Stec EM, Kunapuli P, Holder DJ, Heyse JF, Stulovici B, Ferrer M. 2008. Integrating experimental and analytic approaches to improve data quality in genome-wide RNAi screens. Journal of Biomolecular Screening 13(5): 378-389.

See Also

imageDesign.fn, dualFlashlight.fn, plateWellSeries.fn

Examples

```
data(HTSdataSort)
par( mfrow=c(4, 3) )
imageIntensity.fn(data.df=HTSdataSort, intensityName="log2Intensity",
                  plateName="BARCODE", wellName="WELL_USAGE",
                  rowName="XPOS", colName="YPOS",
                  sampleName="Sample", sourcePlateName="SOBARCODE" )
```

`keepUniqWell.fn` *Unique Well Keeper*

Description

This function keeps rows with unique wells in each plate, without side effects.

Usage

```
keepUniqWell.fn(dataIn.df, colFocus = c("Barcode", "Rowpos", "Colpos"))
```

Arguments

`dataIn.df`	Data from all plates that includes at least three columns for well names, rows, and columns, respectively.
`colFocus`	These columns are the columns that are kept unique and from these the repeats are removed.

Author(s)

Xiaohua Douglas Zhang

Examples

```
data(HTSdata)
dataUniqWell.df = keepUniqWell.fn(dataIn.df=HTSdata, colFocus=c("BARCODE", "XPOS", "YPOS"))
```

`plateWellSeries.fn` *Plate-Well Series Plot*

Description

Function that draws a plate-well series plot for all or part of plates.

Usage

```
plateWellSeries.fn(data.df, intensityName = NA, plateName = "BARCODE", wellName = "WELL_USAGE", rowNa
```

Arguments

data.df	The data for all plates including at least five columns for intensity, plate names, well names, rows, and columns, respectively.
intensityName	Name of the column in data.df indicating intensities in a plate.
plateName	Name of the column in data.df indicating plate names.
wellName	Name of the column in data.df indicating well types.
rowName	Name of the column in data.df indicating row numbers in a plate.
colName	Name of the column in data.df indicating column numbers in a plate.
show.wellTypes	A vector of well types to be shown in the well-series plot.
order.wellTypes	A vector of numbers to indicate the order of well types corresponding to 'show.wellTypes' in the well-series plot.
color.wells	A vector indicating the colors of well types corresponding to 'show.wellTypes' in the well-series plot.
pch.wells	A vector indicating the point types of well types corresponding to 'show.wellTypes' in the well-series plot.
ppf	The number of plates per figure to be shown in the well-series plot.
byRow	Indicates whether the wells in a plate should be shown by row or column.
yRange	Defines the range of the y-axis in the well-series plot.
cex.point	Defines the size of points in the well-series plot.
cex.legend	Defines the size of legend in the well-series plot.
x.legend	Position of legend on x-axis.
y.legend	Position of legend on y-axis.
main	Title of the image.
xlab	The label for the x-axis.
ylab	The label for the y-axis.

Author(s)

Xiaohua Douglas Zhang & Zhaozhi Zhang

References

Zhang XHD, Zhang ZZ. 2013. displayHTS: a R package displaying data and results from high-throughput screening experiments. Bioinformatics (submitted).

Zhang XHD, 2011. Optimal High-Throughput Screening: Practical Experimental Design and Data Analysis for Genome-scale RNAi Research. Cambridge University Press, Cambridge, UK.

Zhang XHD, Yang XC, Chung N, Gates AT, Stec EM, Kunapuli P, Holder DJ, Ferrer M, Espeseth AS. 2006. Robust statistical methods for hit selection in RNA interference high throughput screening experiments. Pharmacogenomics 7 (3) 299-309.

See Also

imageDesign.fn, imageIntensity.fn, dualFlashlight.fn

Examples

```
data(HTSdataSort)
wells = as.character(unique(HTSdataSort[, "WELL_USAGE"])); wells
colors = c("black",  "yellow", "grey", "blue", "skyblue", "green", "red")
orders=c(1, 3, 2, 4, 5, 7, 6)
# by row
par( mfrow=c(2,1) )
plateWellSeries.fn(data.df = HTSdataSort, intensityName="log2Intensity",
                   plateName="BARCODE", wellName="WELL_USAGE",
                   rowName="XPOS", colName="YPOS", show.wellTypes=wells,
                   order.wellTypes=orders, color.wells=colors,
                   pch.wells=rep(1, 7), ppf=6, byRow=TRUE,
                   yRange=NULL, cex.point=0.25,cex.legend=0.3)
# by column
par( mfrow=c(2,1) )
plateWellSeries.fn(data.df = HTSdataSort, intensityName="log2Intensity",
                   plateName="BARCODE", wellName="WELL_USAGE",
                   rowName="XPOS", colName="YPOS", show.wellTypes=wells,
                   order.wellTypes=orders, color.wells=colors,
                   pch.wells=rep(1, 7), ppf=6, byRow= FALSE,
                   yRange=NULL, cex.point=0.25,cex.legend=0.3)
# display hits
data(HTSresults)
condtSample = HTSresults[, "WELL_USAGE"] == "Sample"
condtUp = HTSresults[,"ssmd"] >= 1 & HTSresults[,"mean"] >= log2(1.2)
condtDown = HTSresults[,"ssmd"] <= -1 & HTSresults[,"mean"] <= -log2(1.2)
sum(condtSample & (condtUp | condtDown) )/sum(condtSample)
hit.vec = as.character(HTSresults[, "WELL_USAGE"])
hit.vec[ condtSample & condtUp ] = "up-hit"
hit.vec[ condtSample & condtDown ] = "down-hit"
hit.vec[ condtSample & !condtUp & !condtDown] = "non-hit"
result.df = cbind(HTSresults, "hitResult"=hit.vec)
wells = as.character(unique(result.df[, "hitResult"])); wells
orders = c(1, 3, 4, 6, 7, 8, 9, 2, 5)
colors = c("black",  "green", "yellow", "red",
           "grey", "purple1", "purple2", "lightblue", "purple3")
par(mfrow=c(1,1))
plateWellSeries.fn(data.df = result.df, intensityName="mean",
                   plateName="SOBARCODE", wellName="hitResult",
                   rowName="XPOS", colName="YPOS", show.wellTypes=wells,
                   order.wellTypes=orders, color.wells=colors,
                   pch.wells=rep(1, 7), ppf=6, byRow= FALSE,
                   yRange=NULL, cex.point=0.5,cex.legend=0.55,
                   y.legend=-0.5)
```

Index