

Package 'COMBIA'

October 8, 2018

Encoding UTF-8

Type Package

Title Synergy/Antagonism Analyses of Drug Combinations

Version 1.0.6

Date 2018-10-07

Author Muhammad Kashif

Maintainer Muhammad Kashif <Muhammad.Kashif@medsci.uu.se>

Description A comprehensive synergy/antagonism analyses of drug combinations with quality graphics and data. The analyses can be performed by Bliss independence and Loewe additivity models. 'COMBIA' provides improved statistical analysis and makes only very weak assumption of data variability while calculating bootstrap intervals (BIs). It is based on heteroscedasticity controlled resampling (bootstrapping) and includes a global (omnibus) test. Finally, package shows analyzed data, 2D and 3D plots ready to use in research publications. 'COMBIA' does not require manual data entry. Data can be directly input from wet lab experimental platforms for example fluostar, automated robots etc. One only needs to call a single function, analyzeCOMBO(), to perform all analysis (examples are provided with sample data).

URL <<http://www.impactjournals.com/oncotarget/index.php?journal=oncotarget&page=article&op=view&path%5B%5D=21895&path%5B%5D=69432>> .

Depends hash, lattice, latticeExtra, oro.nifti

Imports gdata, grDevices, stats, utils

License GPL (>= 2)

SystemRequirements perl (>= 5.10.0)

LazyLoad yes

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-10-08 09:50:26 UTC

R topics documented:

analyzeCOMBO	2
applyBliss	4
applyLoewe	5
calculateSi	6
combineDataFromMultipleFiles	7
createUniquePertbs	8
cVCal	9
extractKey	9
extractReplicateValues	10
extractValuesFromRange	11
loeweModel	12
readFile	13
readFluostarPlates	14
readFMCAValues	15
readMacSynergyValues	17
readOtherValues	18
removeOutliers	19
selectPlate	20
synAntPlot	21
synergySignificant	22
Index	23

analyzeCOMBO	<i>This function calculates significant synergy/antagonism according to Bliss or Loewe model and creates scientific publication ready graphs.</i>
--------------	---

Description

This function calculates significant synergy/antagonism according to Bliss or Loewe model and creates scientific publication ready graphs.

Usage

```
analyzeCOMBO(filename, sheet = 1, model, inputFormates, platetype = "384",
  keyposition = 2, selectionkey = "65000", platekey = 7051,
  minThersholdForCVCal = 0.15, minThersholdForCV = 0.3, wells,
  yConcentration, xConcentration, xDrug, yDrug, cellLine,
  survivalFunc = function(x, y, z) { (x - z)/(y - z) }, nBoot = 5000)
```

Arguments

filename	Name of file containing experimental data. For MS Excel files, working version of Perl must be present in the executable search path.
sheet	Optional, sheet number if excel file is used for input.
model	bliss or loewe.

inputFormates	Any of these three formates "fmca", "macsynergy" and "others" are supported. Example is provided with macsynergy format and test data for this example can be found in installation directory ("extdata") of COMBIA. See files FluoOptima_384_2014-03-28test_M and testDataM in directory "extdata" for format details of "fmca" and "macsynergy". "others" can be any other format.
platetype	Optional default is 384. Only 384 and 96 well plates are supported.
keyposition	Optional default is 2. Usefull for automated barcoded data.
selectionkey	Optional default is 65000.
platekey	Optional barcode.
minThersholdForCVCa1	Optional default is 0.15.
minThersholdForCV	Optional default is 0.3.
wells	wells argument should be in triplet form that is 1-Untreated control wells range, 2-empty wells range and 3-case wells range. Thus in example below (see well argument) experiment has four replicates. "l3:l10","m3:m10","b3:k10" is first replicate. Where "l3:l10" is the location of untreated control values in the test-Data.csv, "m3:m10" is the background/ empty well well values and "b3:k10" are values after treatment.
yConcentration	Y drug Concentrations.
xConcentration	X drug Concentrations.
xDrug	X drug name.
yDrug	Y drug name.
cellLine	Cell/Experiment name.
survivalFunc	Optional default is function (x,y,z) (x-z)/(y-z) i.e (treated - background)/ (untreated - background).
nBoot	Optional Number of time to bootstrap default is 5000

Value

Stores and show graph/data of synergy/antagonism analyses

Author(s)

Muhammad kashif

Examples

```
f1 <- system.file("extdata", "testData.csv", package="COMBIA")
wellR <- list(c("l3:l10", "m3:m10", "b3:k10", "l13:l20", "m13:m20", "b13:k20",
              "l23:l30", "m23:m30", "b23:k30", "l33:l40", "m33:m40", "b33:k40") )
mdl <- "bliss"
xConc <- c(0.00, 0.20, 0.39, 0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50)
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)
xDrug <- "A"
yDrug <- "B"
```

```

cellLine <-"Cell"
analyzeCOMBO(filename = c(fl), model = "bliss", inputFormates = "macsynergy",
              wells = wellR, yConcentration = yConc, xConcentration = xConc,
              xDrug = xDrug, yDrug=yDrug, cellLine = cellLine, nBoot=500)

```

applyBliss

Function calculates Bliss Synergy, associated BIs and global BIs

Description

Function calculates Bliss Synergy, associated BIs and global BIs

Usage

```
applyBliss(noOfRows, noOfCols, rawDataPreProcessed, nBoot)
```

Arguments

noOfRows	Number of rows in the experiment
noOfCols	Number of columns in the experiment
rawDataPreProcessed	Data matrix
nBoot	Number of bootstrap

Value

Three lists, first list consists of Bliss Synergy/Antagonism, lower bound of BI and upper bound of BI. 2nd list consists of global BI of Maximun synergistic combiantion and 3rd list consists of global BI of maximum antagonistic combination.

Author(s)

Muhammad kashif

Examples

```

dataFile <- system.file( "extdata", "rawDataPreProcessed.csv", package="COMBIA" )
dataSample <- read.csv(dataFile, header=FALSE )
nR <- 8
nC <- 10
rslt <- applyBliss(nR, nC, as.matrix(dataSample ), 500)

```

applyLoewe	<i>This function calculates Loewe synergy/antagonism and associated BIs</i>
------------	---

Description

This function calculates Loewe synergy/antagonism and associated BIs

Usage

```
applyLoewe(rawDataPreProcessed, xConcentration, yConcentration, nBoot)
```

Arguments

rawDataPreProcessed
Raw preprocessed experimental data

xConcentration X drug concentrations

yConcentration Y drug concentrations

nBoot Number of times to bootstrap

Value

Three lists, first list consisting of Loewe Synergy/Antagonism, lower bound of BI and upper bound of BI. 2nd list consists of global BI for maximum synergy and 3rd list consists of global BI of maximum antagonistic combination.

Author(s)

Muhammad kashif

Examples

```
## Not run:  
dataFile <- system.file("extdata", "rawDataPreProcessed.csv", package="COMBIA")  
dataSample <- read.csv(dataFile, header=FALSE )  
xConc <- c(0.00, 0.20, 0.39, 0.78, 1.56,3.12, 6.25, 12.50, 25.00, 50)  
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)  
noOFBoot <- 500 # a large number is recommended  
rslt <- applyLoewe(as.matrix(dataSample), xConc, yConc, noOFBoot)  
  
## End(Not run)
```

calculateSi	<i>Calculates survival indices (S.Is) for a range of wells (casewells). S.Is for a range of wells are calculated, that range is specified at the third place of wells argument list. This function call the rangemean function to calculate the mean of the range of the specified range. S.I is calculated by (Case well- meanofemptyrange/mean of controlwell- meanofemptyrange). In the wells argument one should provide arguments in the triplet form that is first one is control data range, second one is the empty data range while third one is the control range.</i>
-------------	--

Description

Calculates survival indices (S.Is) for a range of wells (casewells). S.Is for a range of wells are calculated, that range is specified at the third place of wells argument list. This function call the rangemean function to calculate the mean of the range of the specified range. S.I is calculated by (Case well- meanofemptyrange/mean of controlwell- meanofemptyrange). In the wells argument one should provide arguments in the triplet form that is first one is control data range, second one is the empty data range while third one is the control range.

Usage

```
calculateSi(hashedbackplates, platekey, platetype, rowsperexperiment, wells)
```

Arguments

hashedplates	A hash table of picked plates. It is the output of function "selectPlate".
platekey	It is the key of the plate whose S.I is needed to be calculated.
platetype	It is the type of plate (386 and 96).
rowsperexperiment	It is the argument that specifies if the same experiment is repeated and how many times in a plate. If an experiment is repeated twice in adjacent rows then average of its values will be used in the SI calculation.
wells	This argument can take a list of arguments in the triplet form. Where first argument of triplet is the range of control wells, second argument is the range of empty wells while third one is the range of case wells. It is made so that in labs plates layouts can differ greatly. By using this triplet scheme one can handle a number of palte layouts.

Value

A matrix with S.I showing values where they are actually exist on the plate.

Author(s)

Muhammad Kashif

Examples

```
f <- system.file("extdata", "optima.log", package="COMBIA")
fileDF <- readfile(filename = f, separator = "\t", noofrows_skip=0,
                  platetype="384")
Generatedbarcode <- extractKey(keyposition = 2,
                              rawdata = fileDF, numberofrowsperplate = 17,
                              doubleplateexperiment=TRUE)
hashedplates <- selectPlate(rawdata = fileDF,
                            processedbarcode = Generatedbarcode,
                            numberofrowsperplate = 17,
                            selectionkey = "65000",
                            doubleplateexperiment = TRUE )
survivalindecex <- calculateSi(hashedplates = hashedplates,
                              platekey = "7051", platetype = "384", rowsperexperiment=1,
                              wells = c( "c8:h8", "c1:n1", "c3:c7",    "c8:h8", "c1:n1", "c9:c11",
                                           "c8:h8", "c1:n1", "e3:e7",    "c8:h8", "c1:n1", "e9:e11",
                                           "c8:h8", "c1:n1", "g3:g7",    "c8:h8", "c1:n1", "g9:g11")
                              )
```

combineDataFromMultipleFiles

Combine data from multiple files

Description

Combine data from multiple files

Usage

```
combineDataFromMultipleFiles(yConcentration, xConcentration, replNo, file,
                              totalNumberOfReplicates, siReplicates)
```

Arguments

yConcentration Y drug concentrations
xConcentration X drug concentrations
replNo Number of Replicates in all files
file File name
totalNumberOfReplicates Total number of replicates per files
siReplicates data

Value

Combined data of replicate survival indices from multiple experiments

Author(s)

Muhammad kashif

Examples

```
xConc <- c(0.00, 0.20, 0.39, 0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50)
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)
rN <- 4
fN <- 1
trN <- 4
dataFile <- system.file("extdata", "rawDataPreProcessed.csv", package="COMBIA")
dataSample <- read.csv(dataFile, header=FALSE )
replList <- list(vector, 4)
for( i in 1:4)
{ replList[[i]] <- dataSample[i,] }
rslt <- combineDataFromMultipleFiles(list(yConc),
list(xConc), rN,fN,trN, replList )
```

createUniquePertbs *Function to make unique perturbations of the replicates these will be used incase if CV is greater than threshold.*

Description

Function to make unique perturbations of the replicates these will be used incase if CV is greater than threshold.

Usage

```
createUniquePertbs(totalNumberOfReplicates)
```

Arguments

```
totalNumberOfReplicates
    Total replicate number
```

Value

unique possible perturbations

Author(s)

Muhammad kashif

Examples

```
rslt <- createUniquePertbs(5)
```

cVCal	<i>This function calculates CV</i>
-------	------------------------------------

Description

This function calculates CV

Usage

```
cVCal(vals)
```

Arguments

vals	Values
------	--------

Value

cv of input values

Author(s)

Muhammad kashif

Examples

```
mData <- matrix(1:10, 2,5)
rslt <- cVCal(mData)
```

extractKey	<i>Extracts the keyvalues (Barcode) from a dataset, every plate needs barcode. Keyvalues are extracted from the header of the plates at the position specified by keyposition argument.</i>
------------	---

Description

Extracts the keyvalues (Barcode) from a dataset, every plate needs barcode. Keyvalues are extracted from the header of the plates at the position specified by keyposition argument.

Usage

```
extractKey(keyposition, rawdata, numberofrowsperplate, doubleplateexperiment)
```


Arguments

rawDataUnProcessed	A data matrix
wellRanges	Ranges of wells
wellplace	Place of treated (case) well range
simple	TRUE if survival values are already calculated otherwise it is FALSE
excelFormate	True if ranges are in excel formate

Value

Replicate values

Author(s)

Muhammad kashif

Examples

```
dataFile <- system.file("extdata", "testData.csv", package="COMBIA")
rData <- read.csv( dataFile, skip=0, sep=",", nrows=41,
                  fill=TRUE, header=FALSE,
                  blank.lines.skip = FALSE)[,1:13]
wellR= c( "13:110", "m3:m10", "b3:k10", "113:120", "m13:m20", "b13:k20",
          "123:130", "m23:m30", "b23:k30", "133:140", "m33:m40", "b33:k40")
rslt <- extractReplicateValues(rData, wellR, excelFormate=TRUE )
```

extractValuesFromRange

This function extract numerical indices of a given range e-g B2

Description

This function extract numerical indices of a given range e-g B2

Usage

```
extractValuesFromRange(range, excelFormate)
```

Arguments

range	Range e-g B2
excelFormate	TRUE if range is in spreadsheet formate

Value

Number of starting row, ending row, starting column and ending column

Author(s)

Muhammad kashif

Examples

```
rng <- c("B2")
exclF <- TRUE
rslt <- extractValuesFromRange(rng, exclF)
```

loeweModel

This function applies Loewe Model

Description

This function applies Loewe Model

Usage

```
loeweModel(xConcentration, yConcentration, drugYObs_Mean, drugXObs_Mean)
```

Arguments

xConcentration X drug concentrations
yConcentration Y drug concentrations
drugYObs_Mean Mean of y drug observations
drugXObs_Mean Mean of x drug observations

Value

Loewe Model values

Author(s)

Muhammad kashif

Examples

```
xConcentration <- c(0.00,0.20, 0.39, 0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50.0)
yConcentration <- c(128, 64, 32, 16, 8, 4, 2, 0)
drugXObs_Mean <- c(0.9747255, 0.9197924, 0.9520692, 0.9517162, 0.9032701, 0.7892114,
0.6768190, 0.6524227, 0.4561164)
drugYObs_Mean <- rev( c( 0.93, 0.89, 0.73, 0.42, 0.24, 0.21, 0.11) )
rslt <- loeweModel( xConcentration, yConcentration, drugYObs_Mean, drugXObs_Mean)
```

readFile	<i>Reads experimental data from a file. This function reads the data from specified (excel,log, txt etc) file and store it in a data frame.</i>
----------	---

Description

Reads experimental data from a file. This function reads the data from specified (excel,log, txt etc) file and store it in a data frame.

Usage

```
readFile(filename, separator, sheet, noofrows_skip, readplates,
          numberofrowsperplate, platetype)
```

Arguments

filename	Filename.ext.
separator	Any character(, ; ' etc) that is used as a separator in specified file.
sheet	Need to use only when reading excel files. It is the number of the excel sheet to be read in a worksheet.
noofrows_skip	Number of the rows in the file that should be skipped before starting the data reading.
readplates	Number of the plates that you want to read from a set of plates in a file.This parameter can only be used with excel files. Otherwise it will be ignored.
numberofrowsperplate	It is calculated on the basis of type of plates i-e number of rows per plates are 17 for 384 well plates(16 lines from plates + 1 header lines) and 9 for 96 well plates (8 lines from plates + 1 header lines).
platetype	type of plate used i-e 384 or 96 well plate.

Value

Data frame of file data.

Author(s)

Muhammad Kashif

Examples

```
f <- system.file("extdata", "optima.log", package="COMBIA")
fileDF <- readFile(filename = f, separator = "\t", sheet=1, noofrows_skip=0,
readplates=1, numberofrowsperplate=17, platetype="384")
```

`readFluostarPlates` *Read a file and process it to calculate the Survival indeces(S.I). This function calls other functions to complete its task. It reads a file to separate and regenerate the missing platekeys. Checks are performed to keep regenerated missing keyvalues in sync with data. It calculates survival indeces of the provided control wells, where wells should always be in triplet form that is control well range, empty well range and case well range. It can also handle the double plate experiments in which one plate is read twice and only one of them is selected in S.I calculations. Secondly it can also read the data from the file where a plate is read only one time, still it cope with variations if an experiment is repeated twice or many time in adjacent rows in the file.*

Description

Read a file and process it to calculate the Survival indeces(S.I). This function calls other functions to complete its task. It reads a file to separate and regenerate the missing platekeys. Checks are performed to keep regenerated missing keyvalues in sync with data. It calculates survival indeces of the provided control wells, where wells should always be in triplet form that is control well range, empty well range and case well range. It can also handle the double plate experiments in which one plate is read twice and only one of them is selected in S.I calculations. Secondly it can also read the data from the file where a plate is read only one time, still it cope with variations if an experiment is repeated twice or many time in adjacent rows in the file.

Usage

```
readFluostarPlates(filename, separator = ",", noofrows_skip = 0,
  sheet = "1", readplates = 1, platetype, doubleplateexperiment = TRUE,
  keyposition, selectionkey, platekey, rowsperexperiment = 1, wells)
```

Arguments

<code>filename</code>	value of this argument should be path and filename.ext e=g "e:/optima.txt".
<code>separator</code>	is the sepration character within the file assigned to filename.
<code>noofrows_skip</code>	Number of the rows in the file that should be skipped before starting the data reading.
<code>sheet</code>	Need to use only when reading excel files. It is the number of the excel sheet to be read in a worksheet.
<code>readplates</code>	Number of the plates to read from a set of plates from an excel file, This feature is only workable with xls files.
<code>platetype</code>	Two types of plate formates are supported 384 and 96 wells.
<code>doubleplateexperiment</code>	This parameter can have TRUE & FALSE values only. It is set to TRUE when an experiment is read twice.

keyposition	It is the position of key in the header. Currently it is located at the second position but it can be at any position in the header.
selectionkey	value, that will be used during the selection of plate. Current value is 65000.
platekey	barcode of the plate whose wells you want to measure for Survival index
rowsperexperiment	It is the argument that specifies if the same experiment is repeated and how many times in a plate. If an experiment is repeated twice in adjacent rows then average of its values will be used in the SI calculation.
wells	This argument can take a list of arguments in the triplet form. Where first argument of triplet is the range of control wells, second argument is the range of empty wells while third one is the range of case wells. It is made so that in labs plates layouts can differ greatly. By using this triplet scheme one can handel a number of palte layouts. Values should be given in the according to plate range e-g a4:d5 means start from the a(1) row and first column and continue to d(4) row 5th column.

Value

Matrix of S.I.

Author(s)

Muhammad Kashif

Examples

```
f <- system.file("extdata", "optima.log", package = "COMBIA")
platematrix <- readFluostarPlates(filename = f, platetype = "384",
                                keyposition=2, separator= "\t",
                                selectionkey = "65000", platekey = 7051,
wells = c( "c8:h8","c1:n1","c3:c7",      "c8:h8","c1:n1","c9:c11",
           "c8:h8","c1:n1","e3:e7",      "c8:h8","c1:n1","e9:e11",
           "c8:h8","c1:n1","g3:g7",      "c8:h8","c1:n1","g9:g11" )
           )
```

readFMCAValues

Read data from raw FMCA format and clean for outliers

Description

Read data from raw FMCA format and clean for outliers

Usage

```
readFMCAValues(file, platetype, keyposition, selectionkey, platekey, wells,
               minThersholdForCVcal, minThersholdForCV, yConcentration, xConcentration)
```

Arguments

file	Name of file to be read
platetype	384 etc
keyposition	Bar code position
selectionkey	65000
platekey	Barcode
wells	Wells ranges
minThersholdForCVCaI	Thresoldd for data outliears in CV
minThersholdForCV	Thresold of values in CV not to remove
yConcentration	Concentrations of y drug
xConcentration	Concentrations of x drug

Value

Matrix of replicated survival values

Author(s)

Muhammad kashif

Examples

```
f1 <- system.file("extdata","FluoOptima_384_2014-03-28test.txt", package="COMBIA")
wls <- list(c("A11:H11", "A12:H12","A1:H10", "I11:P11", "I12:P12","I1:P10",
            "A23:H23", "A24:H24","A13:H22", "I23:P23", "I24:P24","I13:P22")
          )
pltype <- "384"
keypos <- 2
seleckey <- "65000"
barCode <- 7049
minThersholdForCVCaI <- 0.1
minThersholdForCV <- 0.3
xConc <- c(0.00, 0.20, 0.39,0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50.00)
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)
readFMCAValues(f1, pltype, keypos, seleckey, barCode,
              wls, minThersholdForCVCaI, minThersholdForCV, xConc, yConc )
```

readMacSynergyValues *Read data from macsynergyII formate and clean for outliers*

Description

Read data from macsynergyII formate and clean for outliers

Usage

```
readMacSynergyValues(file, sheet, nrow = 41, wellRangesExcel,
  minThersholdForCVCal, minThersholdForCV, survivalFunc)
```

Arguments

file	Name of fiele to be read
sheet	Sheet Number
nrow	Number of rows in the sheet
wellRangesExcel	TRUE if wells in excel formate
minThersholdForCVCal	Thresoldd for data outliears in CV
minThersholdForCV	Thresold of values in CV not to remove
survivalFunc	<- function (x,y,z) (x-z)/(y-z) # It can be any function

Value

Matrix of replicated values

Author(s)

Muhammad kashif

Examples

```
f1 <- system.file("extdata", "testData.csv", package="COMBIA")
sh <- 1
wellR <- list(c( "13:110", "m3:m10", "b3:k10", "113:120", "m13:m20", "b13:k20",
  "123:130", "m23:m30", "b23:k30", "133:140", "m33:m40", "b33:k40"))
minThersholdForCV <- 0.3
minThersholdForCVCal <- 0.1
survivalFunc <- function (x,y,z) {(x-z)/(y-z)}
rslt <- readMacSynergyValues(f1, sh, nrow=41, wellR,
  minThersholdForCVCal, minThersholdForCV, survivalFunc)
```

readOtherValues	<i>Read data from raw format and clean for outliers</i>
-----------------	---

Description

Read data from raw format and clean for outliers

Usage

```
readOtherValues(file, sheet, rskip = 0, cStart = 1, wellRangesExcel,
  platetype, minThersholdForCVCal, minThersholdForCV, survivalFunc,
  xConcentration, yConcentration)
```

Arguments

file	Name of file to be read
sheet	Sheet
rskip	Number of rows to skip before reading data, default rskip=0
cStart	Number of column to start reading data, default cStart=1
wellRangesExcel	well ranges in excel formate
platetype	384 or 96
minThersholdForCVCal	Thresolld for data outliears in CV
minThersholdForCV	Thresold of values in CV not to remove
survivalFunc	A function to calculate survival values
xConcentration	Concentrations of drug at x-axis
yConcentration	Concentrations of drugs at y-axis

Value

Matrix of survival values of experimental replicates

Author(s)

Muhammad kashif

Examples

```
f1 <- system.file("extdata", "FluoOptima_384_2014-03-28test.csv", package="COMBIA")
wls <- list( c( "K1:K8", "L1:L8","A1:J8",      "K9:K16", "L9:L16","A9:J16",
  "W1:W8", "X1:X8","M1:V8",      "W9:W16", "X9:X16","M9:V16")
  )
sh <- 1
```

```
rskip <- 0
cStart <- 1
pltype <- "384"
minThersholdForCVCal <- 0.1
minThersholdForCV<- 0.3
survivalFunc <- function (x,y,z) {(x-z)/(y-z)}
xConc <- c(0.00, 0.20, 0.39, 0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50.00)
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)
rslt <- readOtherValues(fl, sh, rskip, cStart, wls, pltype, minThersholdForCVCal,
                        minThersholdForCV, survivalFunc, xConc, yConc )
```

removeOutliers	<i>This function Remove Outliers</i>
----------------	--------------------------------------

Description

This function Remove Outliers

Usage

```
removeOutliers(arrangeReplicates, minThersholdForCVCal, minThersholdForCV)
```

Arguments

arrangeReplicates
A data matrix
minThersholdForCVCal
Threshold for value removal in CV
minThersholdForCV
Values to be excluded

Value

Replicate values

Author(s)

Muhammad kashif

Examples

```
dataFile <- system.file("extdata", "rawDataPreProcessed.csv", package="COMBIA")
dataSample <- read.csv(dataFile, header=FALSE )
minThersholdForCV <- 0.3
minThersholdForCVCal <- 0.1
removeOutliers( as.matrix(dataSample ), minThersholdForCV,
                minThersholdForCVCal)
```

selectPlate	<i>Select one of the two read plates and built a hashtable. One plate from each pair of the read plate is selected in case of double plate experiment on the basis of presence of minimum selection key and if none have maxed out values then one with highest mean value is picked.</i>
-------------	---

Description

Select one of the two read plates and built a hashtable. One plate from each pair of the read plate is selected in case of double plate experiment on the basis of presence of minimum selection key and if none have maxed out values then one with highest mean value is picked.

Usage

```
selectPlate(rawdata, processedbarcode, numberofrowsperplate, selectionkey,  
doubleplateexperiment)
```

Arguments

rawdata	An object(dataframe) of rawdata.
processedbarcode	A vector of regenerated missing keyvalues. In this case it is the output of function "extractKey".
numberofrowsperplate	This argument is not needed when you call function "readFluostarPlates". The number of rows depends upon the geometry of the plates. These are 16 in case of 384well paltes.
selectionkey	keyvalue on basis of which a plate is slected from a pair of plates read in double plate experiment.
doubleplateexperiment	This parameter can have TRUE & FALSE values only. It is set to TRUE when an experiment is read twice.

Value

A hashtable of picked plates.

Author(s)

Muhammad Kashif

Examples

```
f <- system.file("extdata", "optima.log", package="COMBIA")
fileDF <- readfile(filename = f, separator = "\t", noofrows_skip=0,
                  platetype = "384")
Generatedbarcode <- Generatedbarcode <- extractKey(keyposition = 2,
            rawdata = fileDF, numberofrowsperplate = 17, doubleplateexperiment = TRUE)
hashedplates <- selectPlate(rawdata = fileDF,
            processedbarcode = Generatedbarcode, numberofrowsperplate=17,
            selectionkey="65000", doubleplateexperiment = TRUE )
```

synAntPlot

*This function plots the synergy analysis 2D and 3D graphs***Description**

This function plots the synergy analysis 2D and 3D graphs

Usage

```
synAntPlot(processedData, xConcentration, yConcentration, xDrug, yDrug,
            cellline)
```

Arguments

processedData	A matrix to plot
xConcentration	X drug concentrations
yConcentration	Y drug concentrations
xDrug	X drug name
yDrug	Y drug name
cellLine	Cell line name

Value

Plot the values

Author(s)

Muhammad kashif

Examples

```
dataFile <- system.file("extdata", "processedData.csv", package="COMBIA")
procData <- read.csv( dataFile, header=FALSE)
xConc <- c(0.00, 0.20, 0.39, 0.78, 1.56, 3.12, 6.25, 12.50, 25.00, 50)
yConc <- c(128, 64, 32, 16, 8, 4, 2, 0)
xD <- "X_Drug"
yD <- "Y_Drug"
c1N <- "myCell"
rs1t <- synAntPlot(as.matrix(procData),xConc,yConc, xD, yD, c1N)
```

synergySignificant *Function calculates significant synergy/antagonism*

Description

Function calculates significant synergy/antagonism

Usage

```
synergySignificant(synergyCalculationLists, noOfRows, noOfCols, xDrug, yDrug,  
  cellLine)
```

Arguments

synergyCalculationLists	List of synergy antagonism calculations
noOfRows	Number of rows
noOfCols	Number of columns
xDrug	Name of drug on x-axis
yDrug	Name of drug on y-axis
cellLine	Cell Line

Value

Processed data

Author(s)

Muhammad kashif

Examples

```
dataFile <- system.file("extdata", "rawDataPreProcessed.csv", package="COMBIA")  
dataSample <- read.csv(dataFile, header=FALSE)  
nR <- 8  
nC <- 10  
rslt <- applyBliss(nR, nC, as.matrix(dataSample ), 100)  
synergySignificant(rslt, nR, nC,"A", "B", "Cell")
```

Index

analyzeCOMBO, [2](#)
applyBliss, [4](#)
applyLoewe, [5](#)

calculateSi, [6](#)
combineDataFromMultipleFiles, [7](#)
createUniquePertbs, [8](#)
cVCal, [9](#)

extractKey, [9](#)
extractReplicateValues, [10](#)
extractValuesFromRange, [11](#)

loeweModel, [12](#)

readFile, [13](#)
readFluostarPlates, [14](#)
readFMCAValues, [15](#)
readMacSynergyValues, [17](#)
readOtherValues, [18](#)
removeOutliers, [19](#)

selectPlate, [20](#)
synAntPlot, [21](#)
synergySignificant, [22](#)