Package 'htpp.pl'

February 6, 2024

Title High-Throughput Phenotypic Profiling Pipeline

Version 0.2

Description
This package provides code for rapid and robust processing of in vitro high-throughput phenotypic profiling (HTPP) assay data. This involves processing across multiple levels of analysis including raw data normalization, cell viability estimation, and concentration-response modeling of several analysis methods defined in Nyffeler et al. 2023 (PMID: 32862757) using the tcplfit2 R package. Results across all levels of the analysis are stored using MongoDB.

Imports data.table (>= 1.14.8), plyr (>= 1.8.8), devtools (>= 2.4.3), dplyr (>= 1.1.3), tidyr (>= 1.3.0), readr (>= 2.1.4), jsonlite (>= 1.8.7), foreach (>= 1.5.2), mongolite (>= 2.7.2),stringr (>= 1.5.0), tictoc (>= 1.2), ggplot2 (>= 3.4.3), tcplfit2 (>= 0.1.5), rlist (>= 0.4.6.2), parallel (>= 3.6.0), doParallel (>= 1.0.17), tibble (>= 3.2.1)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

License MIT + file LICENSE

R topics documented:

axis.breakJN . . . 2
categoryMahalanobisDistances . . . 3
cellViability_plots . . . 4
Cluster.by.Channel . . . 5
ColumnColors . . . 5
concRespPlot_JN . . . 6
correctChemName . . . 7
curveFit_htppCatMah . . . 7
curveFit_htppFeature . . . 8
curveFit_htppGlobalMah . . . 8
curvePlots_htppCatMah . . . 9
curvePlots_htppFeature . . . 9
curvePlots_htppGlobalMah . . . 10

CVanalysis . . . 10
Euclidean_norm_vec . . . 11
findDataCols . . . 11
generate_cvBMC . . . 12
generate_cvTcpl . . . 12
generate_cvWell . . . 13
generate_htppBmc_catMah . . . 13
generate_htppBmc_globalMah . . . 14
generate_htppCatMah . . . 14
generate_htppFeature_htppCategory . . . 15
generate_htppGlobalMah . . . 16
generate_htppNullChems . . . 17
generate_htppPac_catMah . . . 18
generate_htppPac_globalMah . . . 18
generate_htppProfile . . . 19
generate_htppWell . . . 19
generate_htppWellNorm . . . 20
generate_htppWellTrt_htppChem . . . 21
globalMahalanobisDistances . . . 21
GOLetter . . . 22
Level5 . . . 23
maxJN . . . 23
minJN . . . 24
mongoQuery . . . 24
mongoURL . . . 25
nullProbs_catMah . . . 25
outerFences . . . 26
pacPlots_htppCatMah . . . 27
pacPlots_htppGlobalMah . . . 27
PchShape . . . 28
plotCurves . . . 28
pseudoBmcPlots_htppCatMah . . . 29
Raw2Level4 . . . 30
validate_htpp_sampleKey . . . 31
viability_controlPlot_htppWell . . . 32
viability_controlPlot_htppWellNorm . . . 32
Well3Digit . . . 33

Index **34**

`axis.breakJN` *places a break marker at the position "breakpos" in user coordinates on the axis nominated*

Description

places a break marker at the position "breakpos" in user coordinates on the axis nominated

Usage

```
axis.breakJN(
  axis = 1,
  breakpos,
  bgcol = "white",
  breakcol = "black",
  style = c("slash"),
  brw = 0.02,
  Lwd = 1
)
```

Arguments

axis	numeric: which axis the break will be on
breakpos	numeric: where the break will be
bgcol	character string: the background color; white by default
breakcol	character string: the color of the axis break; black by default
style	list of strings: whether the break is a slash or a zigzag
brw	numeric: the break width; 0.02 by default
Lwd	numeric: line width for the plot; 1 by default

Value

The current plot with zigzag axis breaks at the desired place, to better fit data in a visualization

---

categoryMahalanobisDistances

Calculate category mahalanobis distances for pipeline data

Description

Calculate category mahalanobis distances for pipeline data

Usage

```
categoryMahalanobisDistances(
  Level5,
  FeatureList,
  CategoryName,
  coverVariance,
  minObjects,
  SType = "vehicle control",
  mongoUrl
)
```

Arguments

`Level5`	table: A table of well data at "level 5" in the pipeline
`FeatureList`	matrix: The features for each category
`CategoryName`	character string: The category whose variances are being compared
`coverVariance`	numeric: The known variance of the well data
`minObjects`	numeric: The minimum number of expected objects
`SType`	character string: Defines which sample type will be used for data normalization; "vehicle control" by default
`mongoUrl`	character string: the database where the collections are be stored and the required credentials, generated by the mongoURL function

Value

VarianceExplainedList a list of variances with category Mahalanobis distances calculated

---

`cellViability_plots`

This function will generate a plot for each chem_id in the HTPP dataset (for each cell type if applicable). Creates a plot of cell viability data for each chemical id in the dataset in the location specfied by filepath.

Description

This function will generate a plot for each chem_id in the HTPP dataset (for each cell type if applicable). Creates a plot of cell viability data for each chemical id in the dataset in the location specfied by filepath.

Usage

```
cellViability_plots(file_path, study_name, mongoUrl, refChems = TRUE)
```

Arguments

`file_path`	character string: path to directory where HTPP plots will be populated
`study_name`	character string: name of study to be used for plot titles
`mongoUrl`	character string: the database where the collections are be stored and the required credentials, generated by the mongoURL function
`refChems`	boolean: only make viability plots for reference chemicals instead of all chemicals; default in TRUE

Cluster.by.Channel *Orders the columns to be a heatmap axis*

Description

Orders the columns to be a heatmap axis

Usage

```
Cluster.by.Channel(Data)
```

Arguments

Data	table to be reordered

Value

newOrder

ColumnColors *Assigns a color to each column of data by name*

Description

Assigns a color to each column of data by name

Usage

```
ColumnColors(Data, dataform = "matrix")
```

Arguments

Data	dataframe: the data being assigned color
dataform	character string: whether the data is a "matrix" or a "vector"

Value

array of colors matching each of the columns in the data, to distinguish them in plots

`concRespPlot_JN` *Concentration Response Plot*

Description

Concentration Response Plot

Usage

```
concRespPlot_JN(row, ymin = NULL, ymax = NULL)
```

Arguments

`row`	vector: Row containing response data to be plotted
`ymin`	numeric: y axis minimum
`ymax`	numeric: y axis maximum

Value

a log concentration-response plot

Examples

```
conc <- list(.03, .1, .3, 1, 3, 10, 30, 100)
resp <- list(0, .2, .1, .4, .7, .9, .6, 1.2)
row <- list(conc = conc,
            resp = resp,
            bmed = 0,
            cutoff = 0.25,
            onesd = 0.125,
            name = "some chemical",
            assay = "some assay")
res <- tcplfit2::concRespCore(row, conthits = TRUE)
res <- dplyr::mutate(.data = res,
                     la = NA,
                     q = NA,
                     stype = "test sample",
                     endpoint = "test")
Subset <- dplyr::mutate(.data = res,
                        acc = ifelse(is.na(acc), bmd, acc))
Subset2 <- dplyr::filter(.data = Subset,
                         stype %in% c("test sample") & bmd < 100 & hitcall > 0.90)
concRespPlot_JN(Subset2,ymin=-0.5,ymax=2)
```

`correctChemName` *Replaces colons in chemical names with '-' so they will be read in correctly*

Description

Replaces colons in chemical names with '-' so they will be read in correctly

Usage

```
correctChemName(String)
```

Arguments

`String`	character string: chem names to correct

Value

chem_names that fit the correct pattern

Examples

```
correctChemName("test:chemical")
```

`curveFit_htppCatMah` *Category Mahalanobis curve fitting, adds category Mahalanobis fit data to htpp_tcpl collection*

Description

Category Mahalanobis curve fitting, adds category Mahalanobis fit data to htpp_tcpl collection

Usage

```
curveFit_htppCatMah(minObjects, mongoUrl, rerun = FALSE, nThreads = 1)
```

Arguments

`minObjects`	numeric: The minimum number of objects used to filter the dataset for analysis
`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
`rerun`	boolean: rerun = TRUE will drop existing htpp_tcpl collection for global mah values (htpp_tcpl$remove(query=mongoQuery(approach="category")) and reinsert; FALSE by default
`nThreads`	numeric: the number of threads to use for processing; default is 1

`curveFit_htppFeature`

Feature-level curve fitting, adds feature-level fit data to htpp_tcpl collection

---

Description

Feature-level curve fitting, adds feature-level fit data to htpp_tcpl collection

Usage

```
curveFit_htppFeature(minObjects, mongoUrl, rerun = FALSE, nThreads = 1)
```

Arguments

`minObjects`	numeric: The minimum number of objects used to filter the dataset for analysis
`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
`rerun`	boolean: rerun = TRUE will drop existing htpp_tcpl collection for feature (htpp_tcpl$remove(query=r and reinsert; FALSE by default
`nThreads`	• numeric: the number of threads to use for processing; default is 1

`curveFit_htppGlobalMah`

Global Mahalanobis curve fitting, adds global Mahalanobis fit data to htpp_tcpl collection

---

Description

Global Mahalanobis curve fitting, adds global Mahalanobis fit data to htpp_tcpl collection

Usage

```
curveFit_htppGlobalMah(minObjects, mongoUrl, rerun = FALSE)
```

Arguments

`minObjects`	numeric: The minimum number of objects used to filter the dataset for analysis
`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
`rerun`	boolean: rerun = TRUE will drop existing htpp_tcpl collection for global mah values (htpp_tcpl$remove(query=mongoQuery(approach="global", endpoint="global")) and reinsert; FALSE by default

`curvePlots_htppCatMah` *plot category Mahalanobis distances and save the plots to the location specified by file_path.*

Description

plot category Mahalanobis distances and save the plots to the location specified by file_path.

Usage

```
curvePlots_htppCatMah(file_path, study_name, mongoUrl)
```

Arguments

`file_path`	character: file path to where category mah plots will be created
`study_name`	character string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl

Value

a summary table for debugging

`curvePlots_htppFeature` *Feature plotting function, writes plots of feature-level data to the location specified by file_path*

Description

Feature plotting function, writes plots of feature-level data to the location specified by file_path

Usage

```
curvePlots_htppFeature(file_path, study_name, mongoUrl)
```

Arguments

`file_path`	character: file path to where global mah plots will be created
`study_name`	character string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl

`curvePlots_htppGlobalMah` *Plots global mahalanobis distances from htpp_tcpl collection, writes plots of global data to the location specified by file_path*

Description

Plots global mahalanobis distances from htpp_tcpl collection, writes plots of global data to the location specified by file_path

Usage

```
curvePlots_htppGlobalMah(file_path, study_name, mongoUrl)
```

Arguments

`file_path`	character: file path to where global mah plots will be created
`study_name`	character string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
`mongoUrl`	haracter string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl

`CVanalysis` *Reformats the data into a Mongo collection, normalizes it based on the solvent control and finds the percent responder cells. Inputs the cv_well and cv_image_metadata collections.*

Description

Reformats the data into a Mongo collection, normalizes it based on the solvent control and finds the percent responder cells. Inputs the cv_well and cv_image_metadata collections.

Usage

```
CVanalysis(
  InputPath,
  PlateID,
  SType = "vehicle control",
  mongoUrl,
  minNucleiArea = 30,
  maxNucleiArea = 1000,
  minRoundness = 0.5
)
```

Arguments

InputPath	character string: the input path to the Harmony file
PlateID	character string: the plate_id value
SType	character string: Defines which sample type will be used for data normalization; "vehicle control" by default
mongoUrl	character string: The MongoDB host, user, password and database
minNucleiArea	numeric: The minimum area for something flagged as nucleus for QC
maxNucleiArea	numeric: The maximum area for something flagged as nucleus for QC
minRoundness	numeric: The minimum cell roundness for something to be recognized as a cell

---

Euclidean_norm_vec *Calculate the Euclidean norm of a vector*

Description

Calculate the Euclidean norm of a vector

Usage

```
Euclidean_norm_vec(vect)
```

Arguments

vect	Numeric vector: the vector you will take the euclidean norm of

Value

Numeric: the Euclidean norm of the vector

Examples

```
Euclidean_norm_vec(c(1,5,3,4,12))
```

---

findDataCols *finds columns by name, if inv is true, only finds those columns that exist in the data*

Description

finds columns by name, if inv is true, only finds those columns that exist in the data

Usage

```
findDataCols(Table, inv = F, names = F)
```

Arguments

Table	A table of cell-painting data
inv	A boolean, if it's TRUE, findDataCols only finds those columns that exist in the data
names	A boolean, if it's TRUE, return only the column names, if it's FALSe, return the columns

Value

Either the column names, or the columns in Table, based on whether names is TRUE

---

generate_cvBMC *Creates and populates cell viability bmc (cv_bmc) collection in mongo*

Description

Creates and populates cell viability bmc (cv_bmc) collection in mongo

Usage

```
generate_cvBMC(mongoUrl, rerun = FALSE)
```

Arguments

mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	TRUE will drop existing collection and reinsert; FALSE by default

---

generate_cvTcpl *Creates cell viability collection cv_tcpl based on well and chem data*

Description

Creates cell viability collection cv_tcpl based on well and chem data

Usage

```
generate_cvTcpl(cell_viability, mongoUrl, rerun = FALSE)
```

Arguments

cell_viability	boolean: if cell_viability = TRUE, the CVData_all object in line 218 should pull out data from the cv_well collection, otherwise it will pull data from the htpp_well collection
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	boolean: rerun = TRUE will drop existing cv_tcpl collection and reinsert; FALSE by default

generate_cvWell *Create mongo collection for cell viability by well (cv_well) from well treated collection*

Description

Create mongo collection for cell viability by well (cv_well) from well treated collection

Usage

```
generate_cvWell(file_path, mongoUrl, rerun = FALSE)
```

Arguments

file_path	character string: file path to the top level directory of cell viability Harmony files for an HTPP dataset (i.e., the directory above plate-level directories)
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	boolean: rerun = TRUE will drop existing cv_well collection and reinsert; FALSE by default

generate_htppBmc_catMah *Add category Mahalanobis distance information to htpp_bmc collection*

Description

Add category Mahalanobis distance information to htpp_bmc collection

Usage

```
generate_htppBmc_catMah(
  mongoUrl,
  hitCall = 0.95,
  bmc_max = NA,
  bmc_min = 10^0.5,
  rerun = FALSE
)
```

Arguments

mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
hitCall	numeric (between 0-1): Hitcall threshold from tcplfit2 to use for filtering good BMD values; default is 0.95
bmc_max	numeric: The maximum bmc value if bmd > highest tested conc; default is NA

bmc_min	numeric: Defines the denominator for calculating the minimum bmc value for cases where the bmc is less that the lowest tested conc (i.e., minimum tested conc/bmc_min); default is 10^0.5
rerun	boolean: rerun = TRUE will drop existing entries in htpp_bmc for approach = "global" and endpoint = "global", and reinsert; FALSE by default

---

`generate_htppBmc_globalMah` *Create htpp_bmc collection based on htpp_tcpl and adds global mahalanobis distances into htpp_bmc*

Description

Create htpp_bmc collection based on htpp_tcpl and adds global mahalanobis distances into htpp_bmc

Usage

```
generate_htppBmc_globalMah(
  mongoUrl,
  hitCall = 0,
  bmc_max = NA,
  bmc_min = 10^0.5,
  rerun = FALSE
)
```

Arguments

mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
hitCall	numeric (between 0-1): Hitcall threshold from tcplfit2 to use for filtering good BMD values; default is 0 for no hitcall filtering
bmc_max	numeric: The maximum bmc value if bmd > highest tested conc; default is NA
bmc_min	Defines the denominator for calculating the minimum bmc value for cases where the bmc is less that the lowest tested conc (i.e., minimum tested conc/bmc_min); default is 10^0.5
rerun	rerun = TRUE will drop existing entries in htpp_bmc for approach = "global" and endpoint = "global", and reinsert; FALSE by default

---

`generate_htppCatMah` *Create htpp collection htpp_cat_mah for category Mahalanobis distances*

Description

Create htpp collection htpp_cat_mah for category Mahalanobis distances

Usage

```
generate_htppCatMah(
  coverVariance,
  minObjects,
  mongoUrl,
  varianceExplainedPath,
  nThreads = 1,
  rerun = FALSE
)
```

Arguments

coverVariance	numeric: The value of variance explained used to determine the number of eigen features used in analysis
minObjects	numeric: The minimum number of objects used to filter the dataset for analysis
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
varianceExplainedPath	character string: the path where the function will write variance explained meta-data
nThreads	numeric: the number of threads to use for processing; default is 1
rerun	boolean: rerun = TRUE will drop existing htpp_cat_mah collection and reinsert; have FALSE by default

---

generate_htppFeature_htppCategory

Inserts feature and category data into mongo collection htpp_feature and htpp_category

---

Description

Inserts feature and category data into mongo collection htpp_feature and htpp_category

Usage

```
generate_htppFeature_htppCategory(
  inputPath,
  PlateID,
  mongoUrl,
  file_path = "",
  rerun = FALSE
)
```

Arguments

inputPath	character string: Can either be a truncated path, or a full path to a HTPP data file. If it is truncated, the function will rebuild a full path using file_path
PlateID	character string: The PlateID for the plate being analyzed

mongoUrl	character string: The database where the collections will be stored
file_path	character string: The path to where the input file is located
rerun	boolean: Whether to delete and reinsert into both collections; false by default

---

generate_htppGlobalMah *Calculate, plot and record global Mahalanobis distances from mongo data*

Description

Calculate, plot and record global Mahalanobis distances from mongo data

Usage

```
generate_htppGlobalMah(
  coverVariance,
  minObjects,
  plot_file_path,
  study_name,
  mongoUrl,
  rerun = FALSE
)
```

Arguments

coverVariance	numeric: The value of variance explained used to determine the number of eigen features used in analysis
minObjects	numeric: The minimum number of objects used to filter the dataset for analysis
plot_file_path	character string: file path where variance explained plots will be created
study_name	character string: the name of the experiment used to title the plots
mongoUrl	URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	boolean: rerun = TRUE will drop existing cv_well collection and reinsert; FALSE by default

generate_htppNullChems *Create plots for signal strength and add NULL chemicals to htpp_well_norm*

Description

Create plots for signal strength and add NULL chemicals to htpp_well_norm

Usage

```
generate_htppNullChems(
  n_lowest_conc = 2,
  n_cv_active_conc = 6,
  rel_cellCount = 50,
  plot_file_path,
  study_name,
  mongoUrl,
  ConcList = c(100, 10, 1, 0.1, 0.01, 0.001, 1e-04, 1e-05),
  rerun = FALSE
)
```

Arguments

n_lowest_conc	integer: The number of the lowest concentrations in a concentration series for modeling Null chemical data; Default is 2 (i.e., dose_level 1 and 2)
n_cv_active_conc	integer: The number of cell viability active concentrations to be excluded; default is 6 (i.e., exclude chemicals where dose_level >= 6 are cell viability actives)
rel_cellCount	integer: The relative cell count threshold for excluding well data for Null chemical sampling; default is 50 (i.e., exclude wells with rel_cell_count < 50)
plot_file_path	character string: file path where signal strength plots will be created
study_name	character string: the name of the study
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
ConcList	numeric vector: vector of 8 test concentrations to be used for the NULL chemicals. c(100, 10, 1, 0.1, 0.01, 0.001, 0.0001, 0.00001) by default.
rerun	boolean: rerun = TRUE will drop existing cv_well collection and reinsert; FALSE by default

---

`generate_htppPac_catMah`
add category mahalanobis records to htpp_pac

Description

add category mahalanobis records to htpp_pac

Usage

```
generate_htppPac_catMah(mongoUrl, hit_n_conc = 4, rerun = FALSE)
```

Arguments

`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
`hit_n_conc`	numeric: Number of test concentrations needed during curve fitting to determine if a PAC is a hit; default is 4
`rerun`	boolean: rerun = TRUE will drop existing entries in htpp_bmc for approach = "category", and reinsert; FALSE by default

---

`generate_htppPac_globalMah`
generate htpp_pac from htpp_bmc and add global mahalanobis distance records to htpp_pac

---

Description

generate htpp_pac from htpp_bmc and add global mahalanobis distance records to htpp_pac

Usage

```
generate_htppPac_globalMah(mongoUrl, hit_n_conc = 4, rerun = FALSE)
```

Arguments

`mongoUrl`	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
`hit_n_conc`	numeric: Number of test concentrations needed during curve fitting to determine if a PAC is a hit; default is 4
`rerun`	Boolean: TRUE will drop existing entries in htpp_bmc for approach = "global" and endpoint = "global", and reinsert; default is FALSE

generate_htppProfile *create htpp_profile collection*

Description

create htpp_profile collection

Usage

```
generate_htppProfile(n_cells, relative_cellCount, mongoUrl, rerun = FALSE)
```

Arguments

n_cells	numeric: Minimum threshold for the number of cells to keep for filtering
relative_cellCount	numeric: Minimum threshold of the count of relative number of cells to use for filtering
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	boolean: rerun = TRUE will drop existing collection and reinsert; FALSE by default

generate_htppWell *Creates htppWell collections (htpp_well_raw, htpp_well, htpp_image_metadata)*

Description

Creates htppWell collections (htpp_well_raw, htpp_well, htpp_image_metadata)

Usage

```
generate_htppWell(
  file_path,
  mongoUrl,
  Cell_Type,
  CellArea.Limit,
  NucleiArea.Limit,
  SType = "vehicle control",
  n_max = 2000 * 384,
  rerun = FALSE,
  replace = FALSE
)
```

Arguments

file_path	character string: file path to the top level directory of Harmony files for an HTPP dataset (i.e., the directory above plate-level directories)
mongoUrl	character string: The URL of the mongo database holding the collection, with user credentials to access it
Cell_Type	character string or list of strings: the cell type or types being used
CellArea.Limit	dictionary: A dictionary of cells and their corresponding cell area limits, of the form c("celltype" = list(c(lower,upper)))
NucleiArea.Limit	dictionary: A dictionary of cells and their corresponding nuclei area limits, of the form c("celltype" = list(c(lower,upper)))
SType	character string: The sample type used for normalization. Set to "vehicle control" by default
n_max	numeric: The maximum dimensions of the table
rerun	boolean: Whether to drop and replace the collections in the dataframe before inserting
replace	boolean: Whether you want to replace existing records in the mongo database; false by default

---

generate_htppWellNorm

Create http_well_norm, a collection of normalized well data for all plate groups

---

Description

Create http_well_norm, a collection of normalized well data for all plate groups

Usage

```
generate_htppWellNorm(mongoUrl, rerun = FALSE)
```

Arguments

mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
rerun	boolean: rerun = TRUE will drop existing collection and reinsert; have FALSE by default

generate_htppWellTrt_htppChem

Adding chemical id and sample data to htpp_well_trt, htpp_chem

Description

Adding chemical id and sample data to htpp_well_trt, htpp_chem

Usage

```
generate_htppWellTrt_htppChem(
  SampleKey,
  mongoUrl,
  rerun = FALSE,
  replace = TRUE
)
```

Arguments

SampleKey	dataframe: The sample key with chemical info and metadata
mongoUrl	characters string A mongoUrl with credentials to access the database
rerun	boolean: Whether you want to clear the mongo database as you go and refill it; false by default
replace	boolean: Whether you want to replace existing records in the mongo database; false by default

globalMahalanobisDistances

Calculate global Mahalanobis distances for a table

Description

Calculate global Mahalanobis distances for a table

Usage

```
globalMahalanobisDistances(
  Table1,
  coverVariance,
  minObjects,
  SType = "vehicle control",
  url
)
```

Arguments

Table1	table: A table of well data
coverVariance	numeric: The known variance of the well data
minObjects	numeric: Minimum number of cells for plate to pass QC filter
SType	character string: Defines which sample type will be used for data normalization; "vehicle control" by default
url	character string: The MongoDB host, user, password and database

Value

A list of cumulative proportion, rotation matrix and inverse covariance

---

GOLetter *gives one letter abbreviations to attributes*

Description

gives one letter abbreviations to attributes

Usage

```
GOLetter(Vector)
```

Arguments

Vector	a vector of cell attributes

Value

vector of single-letter abbreviations of the attributes listed, to distinguish points on a plot

Examples

```
GOLetter(c("test"))
```

`Level5` *Level 5 analysis on plate data*

Description

Level 5 analysis on plate data

Usage

```
Level5(PlateGroup, SType = "vehicle control", mongoUrl)
```

Arguments

`PlateGroup`	character string: The plate group id
`SType`	character string: What type of plate is used, for QC check if the stype field in the data agrees with it
`mongoUrl`	character string: The MongoDB url of the database with user credentials

Value

Median, nMAD and normalized well data

`maxJN` *calculate the maximum, but return NA is no number is present (pmax doesn't work with summarise)*

Description

calculate the maximum, but return NA is no number is present (pmax doesn't work with summarise)

Usage

```
maxJN(x)
```

Arguments

`x`	a vector one wants the maximum of

Value

The maximum, or NA if there are no nonzero numbers in X

Examples

```
maxJN(c(4,1,7,10,9))
maxJN(c(NA, NA, NA, NA, NA, NA, NA, NA, 0))
```

`minJN` *calculate the minimum, but return NA is no number is present (pmin doesn't work with summarise)*

Description

calculate the minimum, but return NA is no number is present (pmin doesn't work with summarise)

Usage

```
minJN(x)
```

Arguments

`x`	a vector one wants the minimum of

Value

The minimum, or NA if there are no nonzero numbers in X

Examples

```
minJN(c(4,1,7,10,9))
minJN(c(NA, NA, NA, NA, NA, NA, NA, NA, 0))
```

`mongoQuery` *Build a mongo query from an arbitrary set of params or a list*

Description

Build a mongo query from an arbitrary set of params or a list

Usage

```
mongoQuery(...)
```

Arguments

`...`	(any) = Any set of named parameters OR a single named list of all arguments

Value

JSON query ready to pass to mongolite functions

Examples

```
mongoQuery(approach = "global", endpoint = "global")
```

`mongoURL` *Combines credentials into a MongoURL to access a database*

Description

Combines credentials into a MongoURL to access a database

Usage

```
mongoURL(
  host,
  user,
  passwd,
  db,
  authSource = "admin",
  authMechanism = "SCRAM-SHA-256"
)
```

Arguments

`host`	The database host
`user`	The MongoDB username
`passwd`	The password for that database
`db`	The name of the database
`authSource`	The authentication source
`authMechanism`	The authentication mechanism

Value

a MongoDB URL granting read access to the database

Examples

```
mongoURL(host="test", user="readonly", passwd="passwd", db="test_db")
```

`nullProbs_catMah` *Print out Category-level Null Chemical Maximum Hitcall Probabilities*

Description

Print out Category-level Null Chemical Maximum Hitcall Probabilities

Usage

```
nullProbs_catMah(mongoUrl, null_max_conc = 100)
```

Arguments

mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
null_max_conc	integer: Maximum concentration of Null chemicals; default is 100 (uM)

---

outerFences *Tukey Outer fence function*

Description

Tukey Outer fence function

Usage

```
outerFences(input_vector, iqr.factor = 3)
```

Arguments

input_vector	vector: the data whose outer fences you want to calculate
iqr.factor	numeric: the interquartile range factor. 3 by default.

Value

upper and lower fences

Examples

```
outerFences(input_vector= c(113.86844, 108.47126,
125.22345, 115.17092, 123.61978, 112.45098, 128.88594,
100.98654, 103.95449, 109.41060, 114.59485, 107.66969,
101.75302, 100.46038, 103.86191, 110.25791,
113.22980, 101.13067, 112.93010, 105.88312, 98.46702,
92.14626, 97.33307, 117.61402, 110.81441,
106.10610, 110.93701, 115.87897, 116.07416, 104.51375,
106.85793, 117.94644, 111.48693, 122.34972,
103.06462, 127.57024, 120.70566, 98.62746, 110.22989,
150.98643), iqr.factor=3)
```

pacPlots_htppCatMah

Create plots from htpp PAC collection category documents

Description

Create plots from htpp PAC collection category documents

Usage

```
pacPlots_htppCatMah(file_path, study_name, mongoUrl)
```

Arguments

file_path	character: file path to where global mah plots will be created
study_name	character string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl

pacPlots_htppGlobalMah

Plot PACs from global mahalanobis

Description

Plot PACs from global mahalanobis

Usage

```
pacPlots_htppGlobalMah(file_path, study_name, mongoUrl)
```

Arguments

file_path	character: file path to where global mah plots will be created
study_name	haracter string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl

PchShape *PchShape*

Description

PchShape

Usage

```
PchShape(Parameter.Name)
```

Arguments

Parameter.Name	character string: the parameter names being assigned graph shapes

Value

vector of 2-digit base r plots codes for single-letter abbreviations of the attributes listed, to distinguish points on a plot

Examples

```
PchShape("AGP")
```

plotCurves *Plots concentration-response data to visualize Benchmark Dose (BMD)*

Description

Plots concentration-response data to visualize Benchmark Dose (BMD)

Usage

```
plotCurves(
  Subset,
  xLim = NULL,
  TestedRange = NULL,
  plotDatapoints = F,
  plotBMC = F,
  plotDoserange = F,
  plotNoiseband = T,
  Lwd = 1,
  cexAxis = 1,
  yLim = c(-5.5, 100),
  yTicks = NULL,
  yAxisSteps = 25
)
```

Arguments

Subset	dataframe: the subset of the data to be plotted, for instance subset by a given chemical
xLim	numeric: x limit of the plot; null by default.
TestedRange	the dose range tested; null by default.
plotDatapoints	boolean: whether the individual datapoints should be visualized on the plot
plotBMC	boolean: whether the BMC should be highlighted on the plot; false by default
plotDoserange	boolean: whether the doserange should be plotted; false by default
plotNoiseband	boolean: whether the noise band should be plotted; true by default
Lwd	numeric: line width on the plot; 1 by default
cexAxis	numeric: scale of the axis labels; 1 by default
yLim	numeric: y limit of the plot; c(-5.5, 100) by default
yTicks	numeric: the placement of y axis ticks; null by default.
yAxisSteps	the frequency of y axis ticks, if yTicks is undefined

Value

hill plot of the data

---

pseudoBmcPlots_htppCatMah

plot pseudo-biomarker concentration and write the plots to the location specified in file_path

---

Description

plot pseudo-biomarker concentration and write the plots to the location specified in file_path

Usage

```
pseudoBmcPlots_htppCatMah(file_path, study_name, mongoUrl, bmc_min = 10^0.5)
```

Arguments

file_path	character: file path to where global mah plots will be created
study_name	character string: name of study to be used for plot titles, should follow a similar naming convention used in other functions such as "viability_controlPlot_htppWell.R"
mongoUrl	character string: URL to connect to MongoDB for HTPP dataset; can be created using the mongoURL function in htpp.pl
bmc_min	numeric: Defines the denominator for calculating the minimum bmc value for cases where the bmc is less that the lowest tested conc (i.e., minimum tested conc/bmc_min); default is 10^0.5

`Raw2Level4` *Summarize, clean and format the raw data, and store it as a Mongo collection. Run basic QC before other steps*

Description

Summarize, clean and format the raw data, and store it as a Mongo collection. Run basic QC before other steps

Usage

```
Raw2Level4(
  InputPath,
  PlateID,
  mongoUrl,
  Cell_Type = NULL,
  CellArea.Limit = NULL,
  NucleiArea.Limit = NULL,
  n_max = 2000 * 384,
  SType = "vehicle control"
)
```

Arguments

`InputPath`	character string: Path to the Harmony file
`PlateID`	character string: the plate_id for the well plate
`mongoUrl`	character string: The URL of the mongo database holding the collection, with user credentials to access it
`Cell_Type`	character string: the cell line used; null by default.
`CellArea.Limit`	list: The size range for cells examined. If the Cell_Type is "U-2 OS", "MCF7", "A549", "ARPE-19", "HepG2" or "HTB-9", this will autofill with their ranges. Otherwise list(c(0, 99999999)) by default.
`NucleiArea.Limit`	list: The nucleus area range for cells examined. If the Cell_Type is "U-2 OS", "MCF7", "A549", "ARPE-19", "HepG2" or "HTB-9", this will autofill with their ranges. Otherwise list(c(0,99999999)) by default.
`n_max`	numeric: The maximum dimensions of the table
`SType`	character string: The type of plate used, for instance, whether it is control

`validate_htpp_sampleKey` *Reformatting the sample key into a more machine-readable form for later processing and checking it for errors*

Description

Reformatting the sample key into a more machine-readable form for later processing and checking it for errors

Usage

```
validate_htpp_sampleKey(
  SampleKey,
  skipped_tests = c(),
  max_dose_level = 8,
  dataFrame = FALSE
)
```

Arguments

`SampleKey`	File name, path to file, of the sample key file (in csv format) being used
`skipped_tests`	The names of QC tests you want to skip
`max_dose_level`	The maximum dose level used
`dataFrame`	Boolean, if TRUE, will treat SampleTable as data.frame/data.table input and not read in file; default is FALSE

Value

A reformatted sample key if there were no errors, or a list of errors if there were

Examples

```
sample_key <- data.table::fread(
file = system.file("extdata", "example_sampleKey.csv", package = "htpp.pl"),
sep = ",")

validated_sample_key <- validate_htpp_sampleKey(SampleKey = sample_key,
max_dose_level = 8, skipped_tests = "QCV_0", dataFrame = TRUE)
```

`viability_controlPlot_htppWell` *Create plots from htpp_well and write them to the location specified in file_path*

Description

Create plots from htpp_well and write them to the location specified in file_path

Usage

```
viability_controlPlot_htppWell(
  file_path,
  vehicle_control,
  viability_positive_control,
  study_name,
  mongoUrl
)
```

Arguments

`file_path`	character string: The path to where the plots will go.
`vehicle_control`	character string: chem_id for vehicle control samples
`viability_positive_control`	character string: chem_id for viability positive control samples
`study_name`	character string: name of study to be used for plot titles
`mongoUrl`	character string: The URL of the mongo database holding the collection, with user credentials to access it

`viability_controlPlot_htppWellNorm` *Function to plot normalized well values based on vehicle and viability controls and write the plots to the location specified in file_path*

Description

Function to plot normalized well values based on vehicle and viability controls and write the plots to the location specified in file_path

Usage

```
viability_controlPlot_htppWellNorm(
  mongoUrl,
  file_path,
  vehicle_control,
  viability_positive_control,
  study_name
)
```

Arguments

mongoUrl	A string consisting of the databased url and required credentials, generated by the mongoURL function
file_path	A string consisting of the path where the plots will be stored
vehicle_control	A string containing the vehicle chemical (such as dmso) used in this experiment
viability_positive_control	A string containing the viability positive control chemical
study_name	A string containing the name of the study

Well3Digit *Convert 2 digit values to 3 digit in a vector by adding a preceding 0*

Description

Convert 2 digit values to 3 digit in a vector by adding a preceding 0

Usage

```
Well3Digit(Vector)
```

Arguments

Vector	The vector being modified

Value

The vector with all 3 digit values

Examples

```
Well3Digit(c("100","10","30"))
```

Index

axis.breakJN, 2

categoryMahalanobisDistances, 3
cellViability_plots, 4
Cluster.by.Channel, 5
ColumnColors, 5
concRespPlot_JN, 6
correctChemName, 7
curveFit_htppCatMah, 7
curveFit_htppFeature, 8
curveFit_htppGlobalMah, 8
curvePlots_htppCatMah, 9
curvePlots_htppFeature, 9
curvePlots_htppGlobalMah, 10
CVanalysis, 10

Euclidean_norm_vec, 11

findDataCols, 11

generate_cvBMC, 12
generate_cvTcpl, 12
generate_cvWell, 13
generate_htppBmc_catMah, 13
generate_htppBmc_globalMah, 14
generate_htppCatMah, 14
generate_htppFeature_htppCategory, 15
generate_htppGlobalMah, 16
generate_htppNullChems, 17
generate_htppPac_catMah, 18
generate_htppPac_globalMah, 18
generate_htppProfile, 19
generate_htppWell, 19
generate_htppWellNorm, 20
generate_htppWellTrt_htppChem, 21
globalMahalanobisDistances, 21
GOLetter, 22

Level5, 23

maxJN, 23
minJN, 24
mongoQuery, 24
mongoURL, 25

nullProbs_catMah, 25

outerFences, 26

pacPlots_htppCatMah, 27
pacPlots_htppGlobalMah, 27
PchShape, 28
plotCurves, 28
pseudoBmcPlots_htppCatMah, 29

Raw2Level4, 30

validate_htpp_sampleKey, 31
viability_controlPlot_htppWell, 32
viability_controlPlot_htppWellNorm, 32

Well3Digit, 33