Package ‘PKbioanalysis’

May 21, 2025

Type Package

Title Pharmacokinetic Bioanalysis Experiments Design and Exploration

Version 0.3.0

Maintainer Omar Elashkar <`omar.i.elashkar@gmail.com`>

Description Automate pharmacokinetic/pharmacodynamic bioanalytical procedures based on best practices and regulatory recommendations.
The package impose regulatory constrains and sanity checking for common bioanalytical procedures.
Additionally, 'PKbioanalysis' provides a relational infrastructure for plate management and injection sequence.

License AGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.3.0)

Imports dplyr (>= 1.1.3), tidyselect, stringr (>= 1.5.1), ggplot2 (>= 3.5.1), ggforce (>= 0.4.1), tidyr (>= 1.3.0), glue (>= 1.6.2), checkmate, shiny (>= 1.9.1), DBI, duckdb (>= 1.0.0), bslib, bsicons, shinyWidgets, shinyjs, DiagrammeR, units, DT, stats, shinyalert, htmltools, rlang, grDevices, utils, yaml, rhandsontable, methods

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

URL `https://omarashkar.github.io/PKbioanalysis/`

BugReports `https://github.com/OmarAshkar/PKbioanalysis/issues`

Config/testthat/edition 3

NeedsCompilation no

Author Omar Elashkar [aut, cre] (ORCID: <`https://orcid.org/0000-0002-5505-778X`>)

Repository CRAN

Date/Publication 2025-05-21 17:50:01 UTC

Contents

`add_blank` *Add blank to the plate Can be either double blank (DB), CS0IS+ or CS1IS-*

Description

Add blank to the plate Can be either double blank (DB), CS0IS+ or CS1IS-

Usage

```
add_blank(plate, IS = TRUE, analyte = FALSE)
```

Arguments

`plate`	PlateObj object
`IS`	logical. If TRUE, add IS to the well.
`analyte`	logical. If TRUE, add analyte to the well.

Value

PlateObj

`add_cs_curve` *Add calibration curve to the plate*

Description

Add calibration curve to the plate

Usage

```
add_cs_curve(plate, plate_std, rep = 1)
```

Arguments

`plate`	PlateObj
`plate_std`	character
`rep`	numeric. Number of replicates. Default is 1.

Value

PlateObj

Examples

```
plate <- generate_96() |>
 add_cs_curve(c(1, 3, 5, 10, 50, 100, 200))
plot(plate)
```

`add_DB` *Add double blank (DB) to a plate*

Description

Add double blank (DB) to a plate

Usage

```
add_DB(plate)
```

Arguments

`plate`	PlateObj object

Value

PlateObj

Examples

```
plate <- generate_96() |>
add_DB()
```

`add_DQC` *Add dilution quality control (DQC) to the plate*

Description

Add dilution quality control (DQC) to the plate

Usage

```
add_DQC(plate, conc, fac, rep = 5)
```

Arguments

`plate`	PlateObj object
`conc`	numeric. Concentration of the DQC well.
`fac`	numeric. Factor of the DQC well.
`rep`	numeric. Number of replicates. Default is 5. The current implementation does not check ULOQ or LLOQ boundaries.

`add_QC` *Add quality control samples to the plate*

Description

A function to add QCs to plate. This function assumes adherence to ICH guideline M10 on bioanalytical method validation and study sample analysis Geneva, Switzerland (2022). If you are not following this guideline, you can set ‘reg = TRUE‘ to ignore the restrictions.

Usage

```
add_QC(
  plate,
  lqc_conc,
  mqc_conc,
  hqc_conc,
  extra = NULL,
  n_qc = 3,
  qc_serial = TRUE,
  reg = TRUE
)
```

Arguments

`plate`	PlateObj object
`lqc_conc`	low quality control concentration
`mqc_conc`	medium quality control concentration
`hqc_conc`	high quality control concentration
`extra`	numeric vector of extra QC concentrations. Default is NULL.
`n_qc`	number of QC sets. Default is 3
`qc_serial`	logical. If TRUE, QCs are placed serially
`reg`	logical. Indicates if restrictions should not be applied to the QC samples. Default is TRUE

Value

PlateObj

`add_samples` *Add unknown samples to a plate*

Description

Add unknown samples to a plate

Usage

```
add_samples(
  plate,
  samples,
  time = NA,
  conc = NA,
  dil = NA,
  factor = NA,
  dosage = NA,
  prefix = "S",
  vtime = FALSE
)
```

Arguments

`plate`	PlateObj
`samples`	A vector representing samples names. Must be unique.
`time`	A vector representing time points. If vtime = FALSE, time will propagate to all samples.
`conc`	A vector representing concentration. Must be same length as samples.

`dil`	A vector representing dilution factor. Must be same length as samples.
`factor`	A vector representing factor. Must be same length as samples.
`dosage`	A vector representing dosage. Must be same length as samples.
`prefix`	A prefix to be added before samples names. Default is "S"
`vtime`	A logical. If TRUE, time is a vector of sample length as samples. Default is FALSE.

Details

final name will be of form. Prefix-SampleName-Time-Concentration-Factor samples must be a unique vector and did not exist in the plate before. Time is either a vector or a single value. If it is a vector, it will be repeated for each sample. Conc, dil, factor and dosage are either a vector or a single value. If it is a vector, it must be the corrosponding length of samples.

Value

PlateObj

Examples

```
plate <- generate_96() |>
 add_samples(paste0("T", 1:12))
```

`add_samples_c` *Cartesian product of sample factors to a plate*

Description

Cartesian product of sample factors to a plate

Usage

```
add_samples_c(
  plate,
  n_rep,
  time = NA,
  conc = NA,
  factor = NA,
  dosage = NA,
  prefix = "S"
)
```

Arguments

`plate`	PlateObj
`n_rep`	number of samples to be added
`time`	A vector representing time points
`conc`	A vector representing concentration
`factor`	A vector representing factor
`dosage`	A vector representing dosage
`prefix`	A prefix to be added before samples names. Default is "S"

Details

This function is a variation of ‘add_samples()‘ where size of inputs does not matter. The function will automatically create a combination of all sample names with time, concentration and factor. final name will be of form. Prefix-SampleName-Time-Concentration-Factor

Value

PlateObj

`add_suitability` *Add suitability sample to the plate*

Description

Add suitability sample to the plate

Usage

```
add_suitability(plate, conc, label = "suitability")
```

Arguments

`plate`	PlateObj object.
`conc`	numeric. Concentration of the suitability well.
`label`	character. Label for the suitability well. Default is "suitability".

Value

PlateObj

`build_injec_seq` *Create Injection Sequence*

Description

Create Injection Sequence

Usage

```
build_injec_seq(
  plate,
  method,
  repeat_std = 1,
  repeat_qc = 1,
  repeat_analyte = 1,
  blank_after_top_conc = TRUE,
  blank_at_end = TRUE,
  system_suitability = 0,
  blank_every_n = NULL,
  inject_vol,
  descr = "",
  prefix = Sys.Date(),
  suffix = "1",
  tray = 1,
  explore_mode = FALSE,
  conc_df = NULL
)
```

Arguments

`plate`	PlateObj object
`method`	choose method from database
`repeat_std`	number of re-injections for calibration standards. Default is 1.
`repeat_qc`	number of re-injections for QC wells. Default is 1
`repeat_analyte`	number of re-injections for unknown samples. Default is 1
`blank_after_top_conc`	If TRUE, adding blank after high concentrations of standards and QCS.
`blank_at_end`	If True, adding blank at the end of queue.
`system_suitability`	Number of re-injections for suitability vial.
`blank_every_n`	If no QCs, frequency of injecting blanks between analytes.
`inject_vol`	volume of injection in micro liters.
`descr`	Run description.
`prefix`	string at the beginning of the filename. Default is today's date.

`suffix`	string to be added to the end of the filename. Default is "1".
`tray`	Location in sample manager.
`explore_mode`	options either TRUE or FALSE. Default if FALSE.
`conc_df`	data.frame matching compound name to a scaling factor. Maximum 20 compounds allowed.

Details

explore_mode controls if exploratory samples are to be injected. A random sample from each CS and QC group will be sampled along with 1 blank sample.

Value

InjecListObj object

`combine_injec_lists` *Create Sample List with rigorous design*

Description

Create Sample List with rigorous design

Usage

```
combine_injec_lists(
  sample_lists,
  n_equi = 10,
  equi_pos,
  equi_prefix = Sys.Date(),
  equi_suffix = "equi",
  equi_injec_vol = 0.5
)
```

Arguments

`sample_lists`	a list of sample lists
`n_equi`	number of equilibriation injections
`equi_pos`	position of equilibriation injections. For format check details
`equi_prefix`	prefix for equilibriation injections
`equi_suffix`	suffix for equilibriation injections
`equi_injec_vol`	volume of equilibriation injection

Details

The equi_pos format will be Row:Column format. E.g: "A,1"

Value

InjecListObj object

`combine_plates` *Combine plates in MultiPlate object*

Description

Combine plates in MultiPlate object

Usage

```
combine_plates(plates)
```

Arguments

`plates`	list of PlateObj objects

Value

MultiPlate object

`download_sample_list` *Download sample list from database to local spreadsheet with vendor specific format*

Description

Download sample list from database to local spreadsheet with vendor specific format

Usage

```
download_sample_list(sample_list, vendor)
```

Arguments

`sample_list`	dataframe of sample list either from db or from write_injec_seq
`vendor`	currently only 'masslynx', 'masshunter' and 'analyst' are supported

Details

For all current vendors, the exported format will be in csv format, compatible with the respective software.

Value

dataframe

`fill_scheme` *Filling orientation of the plate*

Description

This function sets the filling scheme of the plate. The filling scheme is used to determine the order in which the samples are filled in the plate. The default filling scheme is horizontal, which means that the samples are filled from left to right and top to bottom. The vertical filling scheme means that the samples are filled from top to bottom and left to right.

Usage

```
fill_scheme(
  plate,
  fill = "h",
  tbound = "A",
  bbound = "H",
  lbound = 1,
  rbound = 12
)
```

Arguments

`plate`	PlateObj
`fill`	character. Filling scheme. Either "h" for horizontal, "v" for vertical.
`tbound`	character. Top bound of the filling scheme. Default is "A"
`bbound`	character. Bottom bound of the filling scheme. Default is "H"
`lbound`	numeric. Left bound of the filling scheme. Default is 1
`rbound`	numeric. Right bound of the filling scheme. Default is 12

Value

PlateObj

`generate_96` *Generate 96 Plate Generate a typical 96 well plate. User need to specify the empty rows which a going to be used across the experiment.*

Description

Generate 96 Plate Generate a typical 96 well plate. User need to specify the empty rows which a going to be used across the experiment.

Usage

```
generate_96(descr = "", start_row = "A", start_col = 1)
```

Arguments

`descr`	plate description.
`start_row`	A letter corresponding to empty rows in a 96 well plate. Default is A.
`start_col`	A number indicating a column number to start with, given the start row. Default is 1.

Value

PlateObj

Examples

```
plate <- generate_96()
plot(plate)

plate <- generate_96("calibration", start_row = "C", start_col = 11)
plot(plate)
```

`length,MultiPlate-method`

Length method for MultiPlate

Description

Length method for MultiPlate

Usage

```
## S4 method for signature 'MultiPlate'
length(x)
```

Arguments

`x`	MultiPlate object

Value

number of plates

`make_calibration_study`

Create a calibration study with calibration standards and QCs

---

Description

Create a calibration study with calibration standards and QCs

Usage

```
make_calibration_study(
  plate,
  plate_std,
  lqc_conc = NULL,
  mqc_conc = NULL,
  hqc_conc = NULL,
  n_qc = NULL,
  qc_serial = FALSE,
  n_CS0IS0 = 1,
  n_CS0IS1 = 2,
  n_CS1IS0 = 1
)
```

Arguments

`plate`	PlateObj object
`plate_std`	vector of calibration standards
`lqc_conc`	LQC concentration
`mqc_conc`	MQC concentration
`hqc_conc`	HQC concentration
`n_qc`	number of QC sets
`qc_serial`	logical. If TRUE, QCs are placed serially
`n_CS0IS0`	number of CS0IS0 (double) blanks
`n_CS0IS1`	number of CS0IS1 blanks
`n_CS1IS0`	number of CS1IS0 blanks

Value

PlateObj

`make_metabolic_study` *Create a metabolic study layout*

Description

Create a metabolic study layout

Usage

```
make_metabolic_study(
  cmpds,
  time_points = c(0, 5, 10, 15, 30, 45, 60, 75, 90, 120),
  n_NAD = 3,
  n_noNAD = 2
)
```

Arguments

`cmpds`	vector of compounds, including any standards
`time_points`	vector of time points
`n_NAD`	number of NAD positive samples. Default is 3
`n_noNAD`	number of NAD negative samples. Default is 2

Details

Note that this function does not require plate object. It will create a plate object automatically and return MultiPlate object

Value

MultiPlate object

`plate_app` *bioanalytic_app*

Description

This function creates a shiny app for plate management

Usage

```
plate_app()
```

Value

A shiny app. No default return value. Can return a PlateObj if reuse_plate_button is clicked

---

`plate_metadata` *Set plate description*

Description

Set plate description

Usage

```
plate_metadata(plate, descr)
```

Arguments

`plate`	PlateObj
`descr`	character. Description of the plate

Value

PlateObj

`plot.PlateObj` *Plotting 96 well plate*

Description

Plotting 96 well plate

Usage

```
## S3 method for class 'PlateObj'
plot(
  x,
  color = "conc",
  Instrument = "",
  caption = "",
  label_size = 1,
  transform_dil = FALSE,
  watermark = "auto",
  path = NULL,
  ...
)
```

Arguments

`x`	PlateObj
`color`	character. Coloring variable. Either "conc", "time", "factor", "samples", "dosage"
`Instrument`	A string placed at subtitle
`caption`	A string place at plate caption
`label_size`	numeric. Size of the label. Default is 15
`transform_dil`	logical. If TRUE, transform the dilution factor to the label
`watermark`	character. If "auto", a watermark is added to the plot. If "none", no watermark is added. Default is "auto"
`path`	Default is NULL, if not null, must be a path to save plate image
`...`	additional arguments passed to ggplot2::ggsave

Value

ggplot object

Examples

```
plate <- generate_96("new_plate", "C", 11) |>
  add_blank(IS = FALSE, analyte = FALSE) |>
  add_blank(IS = TRUE, analyte = FALSE) |>
  add_samples(c(
    "RD_per1", "RD_in1", "RD_T30", "RD_T60", "RD_T90", "RD_per2", "RD_in2",
    "EE_in0", "EE_T30", "EE_in30", "EE_T60", "EE_in60", "EE_T90", "EE_in90"
  ))
plot(plate)
```

`plot_design` *Plot the design of the plate*

Description

Plot the design of the plate

Usage

```
plot_design(plate)
```

Arguments

`plate`	PlateObj object

Value

DiagrammeR object

`register_plate` *Register a plate This will save the plate to the database*

Description

Register a plate This will save the plate to the database

Usage

```
register_plate(plate)
```

Arguments

`plate`	PlateObj object or MultiPlate object

Value

PlateObj object or list of PlateObj objects

`write_injec_seq` *Write injection sequence to database*

Description

Write injection sequence to database

Usage

```
write_injec_seq(injec_seq)
```

Arguments

`injec_seq`	InjecListObj object

Value

dataframe

`[[,MultiPlate-method` *Subsetting method for MultiPlate*

Description

Subsetting method for MultiPlate

Usage

```
## S4 method for signature 'MultiPlate'
x[[i, j, ...]]
```

Arguments

`x`	MultiPlate object
`i`	index
`j`	index
`...`	additional arguments

Value

PlateObj object

Index