Package 'exeval'

June 10, 2026

Type Package

Language en-US

Title External Evaluation of Population Pharmacokinetic-Pharmacodynamic (popPKPD) Models

Version 0.0.1

Description Provides tools to automate external pharmacokinetic model evaluation workflows, including Bayesian forecasting, predictive performance metrics, diagnostic plotting, and automated reporting.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1)

Imports mapbayr, dplyr, ggplot2, ggpubr, mrgsolve (>= 1.0.8), scales, stats, methods, rlang

Suggests knitr, rmarkdown,

URL https://github.com/Martin-Umpierrez/exeval

BugReports https://github.com/Martin-Umpierrez/exeval/issues

RoxygenNote 7.3.3

NeedsCompilation no

Author Manuel Ibarra [aut], Martin Umpierrez [aut, cre], Ignacio Alvarez Castro [aut] (ORCID: <https://orcid.org/0000-0003-1633-2432>), Nicolas Schmidt [aut] (ORCID: <https://orcid.org/0000-0001-5083-5792>)

Maintainer Martin Umpierrez <mumpierrez@fq.edu.uy>

Repository CRAN

Date/Publication 2026-06-10 08:10:14 UTC

Contents

`combine_metrics` *Combine evaluation metrics across multiple models*

Description

Combines model performance metrics from multiple externally evaluated models into a single data frame and identifies the top-performing models according to a selected metric and ranking criterion.

Usage

```
combine_metrics(
  models,
  metric = "MAIPE",
  top_n = 2,
  occ_eval = 2,
  rank_criteria = "min"
)
```

Arguments

`models`	A list of model evaluation results. Each element must be a named list containing: • `model_name`: Character string identifying the model. • `metrics_list`: Output object containing `metrics_means`, typically generated by `metrics_occ()`.
`metric`	Character string specifying the metric used for ranking models (e.g., `"MAIPE"`, `"IF20"`, `"IF30"`, `"rBIAS"`).
`top_n`	Integer. Number of top-performing models to retain.
`occ_eval`	Numeric or character value specifying the evaluation occasion (`OCC`) used for ranking. Default is `2`.
`rank_criteria`	Character string specifying the ranking rule: • `"min"`: selects models with the lowest metric values (e.g., `MAIPE`). • `"max"`: selects models with the highest metric values (e.g., `IF30`). • `"abs"`: selects models with the smallest absolute metric values (e.g., `rBIAS`).

Details

All models must have been evaluated using the same external evaluation strategy (`eval_type`), otherwise the function stops with an error.

Top-performing models are identified internally according to the selected metric and ranking criterion.

Value

A named list containing:

cmetrics Combined data frame containing metrics for all input models, with an additional `Model` column.

topmodelspd Data frame containing the top-performing models according to the selected ranking criteria.

See Also

`metrics_occ()`

Examples

```
#' # Fake  metrics
generate_fake_metrics <- function(n_occasions = 3) {
data.frame(
 OCC = rep(1:n_occasions),  # Simula varias ocasiones
 rBIAS = stats::rnorm(n_occasions, mean = 0, sd = 10),
 rBIAS_lower = stats::rnorm(n_occasions, mean = -5, sd = 5),
 rBIAS_upper = stats::rnorm(n_occasions, mean = 5, sd = 5),
 MAIPE = stats::runif(n_occasions, min = 10, max = 50),
 IF20 = stats::runif(n_occasions, min = 20, max = 80),
 IF30 = stats::runif(n_occasions, min = 30, max = 90)
)
}
simulation1 <- list(metrics_means = generate_fake_metrics())
simulation2 <- list(metrics_means = generate_fake_metrics())
# List of models
models_list <- list(
list(model_name = "Test_Model1", metrics_list = simulation1),
list(model_name = "Test_Model2", metrics_list = simulation2)
)

combined_results <- combine_metrics(models_list)
print(combined_results)
```

combine_metric_plot *Plot combined model performance metrics*

Description

Generates comparative visualizations of evaluation metrics across multiple externally evaluated models combined with `combine_metrics()`.

Usage

```
combine_metric_plot(
  cmetrics,
  type = c("bias_barplot", "MAIPE_barplot", "IF20_plot", "IF30_plot")
)
```

Arguments

`cmetrics`	Named list returned by `combine_metrics()`, containing combined model performance metrics.
`type`	Character string specifying the plot type to generate. Available options are: • `"bias_barplot"`: bar plot of relative bias (`rBIAS`) with confidence intervals. • `"MAIPE_barplot"`: bar plot of mean absolute individual prediction error (`MAIPE`). • `"IF20_plot"`: bar plot of `IF20` values with reference threshold. • `"IF30_plot"`: bar plot of `IF30` values with reference threshold.

Details

This function is intended for model comparison workflows, allowing visual inspection of predictive performance across evaluation occasions..

Metrics are displayed separately for each evaluation occasion (`OCC`) using faceted plots, enabling direct visual comparison between candidate models.

Value

A `ggplot2` object.

See Also

`combine_metrics()`, `plot.EvalPPK()`

Examples

```
# Fake  metrics
generate_fake_metrics <- function(n_occasions = 3) {
data.frame(
 OCC = rep(1:n_occasions),  # Simula varias ocasiones
 rBIAS = stats::rnorm(n_occasions, mean = 0, sd = 10),
 rBIAS_lower = stats::rnorm(n_occasions, mean = -5, sd = 5),
 rBIAS_upper = stats::rnorm(n_occasions, mean = 5, sd = 5),
 MAIPE = stats::runif(n_occasions, min = 10, max = 50),
 IF20 = stats::runif(n_occasions, min = 20, max = 80),
 IF30 = stats::runif(n_occasions, min = 30, max = 90)
)
}

simulation1 <- list(metrics_means = generate_fake_metrics())
simulation2 <- list(metrics_means = generate_fake_metrics())
# List of models
models_list <- list(
list(model_name = "Test_Model1", metrics_list = simulation1),
list(model_name = "Test_Model2", metrics_list = simulation2)
)
combined_results <- combine_metrics(models_list)

# plot results
combine_metric_plot(combined_results, type = 'bias_barplot')
```

`exeval_models` *Built-in population PK/PKPD models*

Description

Curated population pharmacokinetic (PK) and pharmacokinetic-pharmacodynamic (PKPD) models included in the package for external evaluation workflows.

Usage

```
exeval_models
```

Format

A data frame with 6 variables:

Label Unique model identifier used to reference the model within the package.

Drug Drug associated with the model.

Author First author of the original publication.

Year Publication year.

Ref Reference title or citation for the original model publication.

Model_code Model code stored as a character string in **mrgsolve** format.

Details

These models can be used directly in `exeval_ppk()` by supplying the corresponding `Label` as the `model` argument.

`exeval_ppk` *External evaluation workflow for population PK, PKPD models*

Description

Runs the complete external evaluation workflow for population pharmacokinetic (popPK) or pharmacokinetic-pharmacodynamic (PKPD) models, including MAP estimation, posterior model updating, simulation, and prediction error metric calculation.

Prints a formatted representation of an `EvalPPK` object, including dataset characteristics, evaluation settings, and performance metrics.

Prints a formatted representation of a summary generated from an `EvalPPK` object, including metadata, applied summary settings, global performance metrics, fit distribution, and poorly fitted individuals.

Generates a structured summary of an `EvalPPK` object, including global performance metrics, fit quality classification, and identification of poorly fitted individuals.

Generates visualization plots for external evaluation results stored in an `EvalPPK` object, including prediction error metrics, fit quality distributions, and forecasting performance summaries.

Usage

```
exeval_ppk(
  model,
  data,
  model_name = NULL,
  drug_name = NULL,
  tool = "mapbayr",
  check_compile = TRUE,
  num_occ = NULL,
  num_ids = NULL,
  sampling = TRUE,
  occ_ref = NULL,
  evaluation_type = c("sequential_updating", "stepwise_updating",
    "sequential_reference_updating", "backward_reference_updating"),
  method = c("L-BFGS-B", "newuoa"),
  assessment = c("a_priori", "Bayesian_forecasting", "Complete"),
  verbose = FALSE
)

## S3 method for class 'EvalPPK'
print(x, ...)
```

```
## S3 method for class 'summary.EvalPPK'
print(x, ...)

## S3 method for class 'EvalPPK'
summary(
  object,
  occ = NULL,
  by_occ = TRUE,
  poor_threshold = 50,
  top_n = 10,
  ...
)

## S3 method for class 'EvalPPK'
plot(
  x,
  type = c("bias_barplot", "bias_pointrange", "MAIPE_barplot", "bias_boxplot",
   "bias_violin", "bias_dotplot", "bias_density", "IF20_plot", "IF30_plot", "IF_plot",
    "error_plot", "fit_class", "fit_histogram"),
  occ = NULL,
  signed = FALSE,
  ...
)
```

Arguments

`model`	Population PK model provided as one of the following: • a character string containing **mrgsolve** model code, • a compiled `mrgsolve::mrgmod` object, • or a model label matching an entry in the internal `exeval_models` database. If model code is supplied as a character string, `model_name` must also be provided.
`data`	Data frame containing the external evaluation dataset. Must include at least `ID`, `OCC`, and `CMT`. See `prepare_data()` for expected input formatting.
`model_name`	Character string. Name used when compiling the model with `mrgsolve::mcode()`. Required only when `model` is provided as character code.
`drug_name`	Character string used for reporting purposes only.
`tool`	Character string specifying the estimation backend. Currently only `"mapbayr"` is supported.
`check_compile`	Logical. If `TRUE`, checks model compatibility with **mapbayr** before estimation.
`num_occ`	Integer. Maximum number of occasions to include in the evaluation. If `NULL`, all available occasions are used.
`num_ids`	Integer. Number of subjects to include. If `NULL`, all unique subjects are used.
`sampling`	Logical. If `TRUE`, subjects are randomly sampled when `num_ids` is specified. Otherwise, the first subjects are used.

occ_ref — Integer. Reference occasion used for "sequential_reference_updating" and "backward_reference_updating" evaluation strategies.

evaluation_type — Character string specifying the evaluation strategy:

- "sequential_updating": cumulative MAP updating across occasions.
- "stepwise_updating": independent MAP estimation per occasion.
- "sequential_reference_updating": cumulative MAP updating up to a reference occasion.
- "backward_reference_updating": backward updating from a reference occasion.

method — Character string specifying the optimization algorithm passed to mapbayr::mapbayest(). Supported options are "L-BFGS-B" and "newuoa".

assessment — Character string specifying the simulation strategy. Available options are:

- "a_priori": simulates predictions using the population model without individual posterior information.
- "Bayesian_forecasting": simulates predictions using individualized posterior parameter estimates obtained from MAP estimation.
- "Complete": performs both a priori and Bayesian forecasting simulations.

verbose — Logical. If TRUE, progress messages are printed during execution.

x — An object of class EvalPPK.

... — Additional arguments passed to or from other methods.

object — An object of class EvalPPK.

occ — Optional numeric occasion (OCC) to filter the plot. If NULL (default), all available occasions are included.

by_occ — Logical. If TRUE, summaries are stratified by occasion (OCC). Cannot be used together with occ.

poor_threshold — Numeric threshold defining poor fit based on absolute individual prediction error (|IPE|). Default is 50.

top_n — Integer. Number of poorly fitted individuals to report. Default is 10.

type — Character string specifying the type of plot to generate. Available options are:

- "bias_barplot": bar plot of relative bias (rBIAS) with confidence intervals.
- "bias_pointrange": point-range plot of relative bias (rBIAS) with confidence intervals.
- "MAIPE_barplot": bar plot of mean absolute individual prediction error (MAIPE) by occasion.
- "bias_boxplot": boxplot of individual prediction errors (IPE) by occasion.
- "bias_violin": violin plot of individual prediction errors (IPE) by occasion.
- "bias_dotplot": jittered dot plot of individual prediction errors (IPE) by occasion.

- `"bias_density"`: density plot of individual prediction errors across occasions.
- `"IF20_plot"`: bar plot of `IF20` values with reference threshold.
- `"IF30_plot"`: bar plot of `IF30` values with reference threshold.
- `"IF_plot"`: combined visualization of both `IF20` and `IF30`.
- `"error_plot"`: stacked bar plot showing the proportion of observations within predefined prediction error categories.
- `"fit_class"`: bar plot showing the distribution of fit quality categories.
- `"fit_histogram"`: histogram of individual prediction error values.

`signed` Logical. Only used when `type = "fit_histogram"`. If `TRUE`, signed individual prediction errors are plotted. If `FALSE` (default), absolute individual prediction errors are used.

Details

This function serves as the main high-level interface for the **exeval** workflow.

This function executes the complete external evaluation workflow:

1. MAP estimation via `run_MAP_estimations()`
2. posterior model updating via `update_map_models()`
3. PK simulations via `run_pk_simulations()`
4. prediction error metric calculation via `metrics_occ()`

The returned object is an `EvalPPK` object containing all intermediate results and summary metadata.

Summary outputs can be generated across all occasions, for a specific occasion, or stratified by occasion.

This method provides visualization tools for assessing predictive performance of external model evaluations, including bias, precision, forecasting success, and fit quality classification.

Value

An object of class `EvalPPK` containing:

metrics Prediction error metrics returned by `metrics_occ()`.

estimates MAP estimation results returned by `run_MAP_estimations()`.

updates Posterior individualized models returned by `update_map_models()`.

simulations Simulation outputs returned by `run_pk_simulations()`.

Additional workflow metadata are stored as object attributes.

An object of class `summary.EvalPPK` containing:

metadata Evaluation metadata inherited from the original `EvalPPK` object.

global_metrics Summary performance metrics across all observations or stratified by occasion.

fit_distribution Distribution of fit quality categories based on absolute prediction error.

poor_fit_ids Table of individuals exceeding the selected poor-fit threshold.

A `ggplot2` object, except for `"IF_plot"`, which returns a combined plot object generated with **ggpubr**.

See Also

run_MAP_estimations(), update_map_models(), run_pk_simulations(), metrics_occ()

Examples

```
data("tacrolimus_pk1_kidney", package = "exeval")
data("model_tacHAN2011", package = "exeval")

dd <- tacrolimus_pk1_kidney |> subset(ID < 6)

res <- exeval_ppk(model="TAC_Han2011",
                  data = dd,
                  evaluation_type= "sequential_updating",
                  assessment='Bayesian_forecasting' )

print(res) # Print the results
```

---

metrics_occ *Compute external evaluation performance metrics*

Description

Computes predictive performance metrics from simulation outputs generated during external model evaluation.

Usage

```
metrics_occ(
  simulations,
  assessment = c("a_priori", "Bayesian_forecasting", "Complete"),
  tool = "mapbayr",
  ...
)
```

Arguments

simulations	Named list returned by run_pk_simulations() containing simulation outputs and treatment/event data.
assessment	Character string specifying the prediction strategy used to generate the simulations. Available options are: • "a_priori": evaluates predictions generated from the population model. • "Bayesian_forecasting": evaluates predictions generated from individualized posterior models. • "Complete": evaluates both a priori and Bayesian forecasting predictions.
tool	Character string specifying the simulation backend. Currently only "mapbayr" is supported
...	Additional arguments (currently unused).

Details

This function compares simulated predictions with observed concentrations and calculates individual- and occasion-level prediction error metrics.

Individual predictions are matched with observed concentrations using subject identifier (`ID`), occasion (`OCC`), and observation time (`TIME`).

The following metrics are calculated:

- `IPE`: individual prediction error (%)
- `APE`: absolute prediction error (%)
- `rRMSE`: relative root mean squared error (%)
- `rBIAS`: relative bias (%)
- `MAIPE`: mean absolute individual prediction error (%)
- `IF20`: percentage of predictions within 20\
- `IF30`: percentage of predictions within 30\

Individual observations are additionally classified into fit quality categories (`Excellent`, `Acceptable`, `Poor`, `Very Poor`) based on absolute prediction error.

Value

A named list containing:

metrics Data frame containing individual prediction errors and fit classifications for each subject, occasion, and observation time.

metrics_means Data frame containing summary performance metrics aggregated by occasion.

See Also

`run_pk_simulations()`, `plot.EvalPPK()`, `summary.EvalPPK()`

Examples

```
data("exeval_models", package = "exeval")
data("tacrolimus_pk1_kidney", package = "exeval")

dd <- tacrolimus_pk1_kidney |> subset(ID < 6)

fit <- run_MAP_estimations(
  model = exeval_models$Model_code[[2]],
  model_name = "TAC_Zuo2013",
  data = dd,
  evaluation_type = "sequential_updating"
)

post <- update_map_models(
  map_results = fit,
  evaluation_type = "sequential_updating"
)
```

```
sim <- run_pk_simulations(
  individual_model = post,
  map_results = fit,
  assessment = "Complete"
)

mm <- metrics_occ(
  simulations = sim,
  assessment = "Complete"
)
```

`prepare_data` *Prepare input data for exeval*

Description

Renames user-defined dataset columns to the standardized naming convention used internally by **exeval**.

Usage

```
prepare_data(
  data,
  name_id = NULL,
  name_time = NULL,
  name_occ = NULL,
  name_date = NULL,
  name_cmt = NULL,
  name_dv = NULL,
  name_mdv = NULL,
  name_amt = NULL,
  name_evid = NULL,
  name_ss = NULL,
  name_dur = NULL,
  name_lag = NULL,
  name_rate = NULL,
  name_ii = NULL,
  name_addl = NULL
)
```

Arguments

`data`	A data frame containing the input dataset.
`name_id`	Character. Name of the column containing subject IDs.
`name_time`	Character string. Name of the sampling or event time column.

name_occ	Optional character. Name of the occasion column.
name_date	Optional character. Name of the date column.
name_cmt	Optional character. Name of the compartment column.
name_dv	Optional character string. Name of the dependent variable (observed concentration/response) column.
name_mdv	Optional character string. Name of the missing dependent variable indicator column.
name_amt	Optional character. Name of the dose amount column.
name_evid	Optional character. Name of the event ID column.
name_ss	Optional character string. Name of the steady-state indicator column.
name_dur	Optional character string. Name of the infusion duration column.'
name_lag	Optional character string. Name of the lag time column.
name_rate	Optional character string. Name of the infusion rate column.
name_ii	Optional character string. Name of the interdose interval column.
name_addl	Optional character string. Name of the additional doses column.

Details

This helper function allows external datasets with arbitrary column names to be reformatted for compatibility with the external evaluation workflow.

At minimum, `ID` and `TIME` mappings must be provided.

Additional columns can be optionally mapped depending on the analysis workflow and model requirements.

Value

A data frame with standardized column names compatible with **exeval**.

See Also

`exeval_ppk()`

Examples

```
df <- data.frame(
  patient = c(1, 1, 2),
  time = c(0, 12, 24),
  conc = c(NA, 8.4, 6.1),
  visit = c(1, 1, 2)
)

df_std <- prepare_data(
  data = df,
  name_id = "patient",
  name_time = "time",
  name_dv = "conc",
```

```
  name_occ = "visit"
)

head(df_std)
```

`run_MAP_estimations` *Run MAP Bayesian Estimation for External Model Evaluation*

Description

Performs Maximum A Posteriori (MAP) Bayesian estimation using **mapbayr** for external evaluation of pharmacokinetic models across multiple dosing occasions.

Usage

```
run_MAP_estimations(
  model,
  model_name = NULL,
  tool = "mapbayr",
  check_compile = TRUE,
  data,
  num_occ = NULL,
  num_ids = NULL,
  sampling = TRUE,
  occ_ref = NULL,
  evaluation_type = c("sequential_updating", "stepwise_updating",
    "sequential_reference_updating", "backward_reference_updating"),
  method = c("L-BFGS-B", "newuoa")
)
```

Arguments

`model`	Population PK model, provided either as: • A character string containing the pharmacokinetic model code written in `mrgsolve` format. • A pre-compiled `mrgmod` object (S3 class from `mrgsolve`). If a character string is provided, `model_name` must also be specified.
`model_name`	Character string. Name used when compiling the model with `mrgsolve::mcode()`. Required only when `model` is provided as character model code.
`tool`	Character string. Estimation engine to use. Currently only `"mapbayr"` is supported.
`check_compile`	Logical. If `TRUE`, validates model compatibility with **mapbayr** before estimation.
`data`	Data frame containing external evaluation data. Must include at least `ID`, `OCC`, and `CMT`. See `prepare_data()` for expected formatting and preprocessing.

`num_occ` Integer. Maximum number of occasions to include in the analysis. If `NULL`, all available occasions in the data are used.

`num_ids` Integer. Number of subjects to include. If `NULL`, all unique subjects are used.

`sampling` Logical. If `TRUE`, subjects are randomly sampled when `num_ids` is specified. Otherwise, the first `num_ids` subjects are selected.

`occ_ref` Integer. Reference occasion used for reference-based evaluation strategies. Required when `evaluation_type` is `"sequential_reference_updating"` or `"backward_reference_upd` where MAP estimation is performed relative to this occasion.

`evaluation_type`
Character string specifying the evaluation strategy. Available options are:

- `"sequential_updating"`: performs MAP estimation using all observations accumulated up to each occasion.
- `"stepwise_updating"`: performs MAP estimation using observations from each occasion independently.
- `"sequential_reference_updating"`: performs MAP estimation using cumulative observations up to the reference occasion defined by `occ_ref`.
- `"backward_reference_updating"`: performs MAP estimation by sequentially moving backward from the reference occasion defined by `occ_ref`.

`method` Character string specifying the optimization algorithm passed to `mapbayr::mapbayest()` for MAP estimation. Supported options are `"L-BFGS-B"` and `"newuoa"`.

Details

The population model can be provided either as:

- a compiled `mrgsolve::mrgmod` object, or
- a character string containing **mrgsolve** model code.

When model code is supplied as a character string, the model is compiled internally using `mrgsolve::mcode()`. In this case, a model name must be provided via `model_name`.

The evaluation strategy defines which observations are used to inform each MAP estimation:

- `"sequential_updating"`: cumulative observations up to each occasion (e.g., OCC1, OCC1+2, OCC1+2+3).
- `"stepwise_updating"`: observations from each occasion treated independently.
- `"sequential_reference_updating"`: cumulative observations up to the reference occasion `occ_ref`.
- `"backward_reference_updating"`: sequential backward updating from `occ_ref`.

Value

A named list containing:

data_by_occ List of input datasets partitioned according to the selected evaluation strategy, where each element contains the observations used for a specific MAP estimation.

treatments_by_occ List of treatment/event datasets grouped by occasion and subject, used for posterior predictive simulations.

apriori_treatments List of treatment/event datasets used for a priori predictive simulations.

map_estimations List of MAP estimation objects returned by `mapbayr::mapbayest()` for each evaluation subset.

eval_type Character string indicating the selected evaluation strategy.

pop_model Compiled population model (`mrgmod`) used for estimation.

See Also

`mapbayr::mapbayest()`, `mrgsolve::mcode()`

Examples

```
data("exeval_models", package = "exeval")
data("tacrolimus_pk1_kidney", package = "exeval")

dd <- tacrolimus_pk1_kidney |> subset(ID < 6)

fit <- run_MAP_estimations(
  model = exeval_models$Model_code[[2]],
  model_name = "TAC_Zuo2013",
  data = dd,
  evaluation_type = "sequential_updating"
)
```

---

run_pk_simulations *Run PK simulations for external model evaluation*

Description

Simulates concentration-time profiles for external model evaluation using population predictions (a priori), individualized posterior predictions (Bayesian forecasting), or both, depending on the selected simulation strategy.

Usage

```
run_pk_simulations(
  individual_model,
  map_results,
  assessment = c("a_priori", "Bayesian_forecasting", "Complete"),
  seed = NULL,
  verbose = FALSE
)
```

Arguments

`individual_model`
: Named list returned by `update_map_models()` containing individualized posterior models used for Bayesian forecasting. Required when `assessment` includes Bayesian forecasting.

`map_results`
: Named list returned by `run_MAP_estimations()` containing treatment/event datasets, evaluation metadata, and the population model required for simulation.

`assessment`
: Character string specifying the simulation strategy. Available options are:

- `"a_priori"`: simulates concentration-time profiles using the population model only.
- `"Bayesian_forecasting"`: simulates concentration-time profiles using individualized posterior models.
- `"Complete"`: performs both a priori and Bayesian forecasting simulations.

`seed`
: Optional integer used to set the random number generator seed for reproducible a priori simulations. If `NULL` (default), the current random number generator state is used.

`verbose`
: Logical. If `TRUE`, progress messages are printed during execution. If `FALSE`, simulation errors are returned as warnings..

Details

This function performs pharmacokinetic simulations at the observation times available in the external evaluation dataset.

Depending on the selected `assessment`, simulations are performed using:

- the population model for a priori predictions,
- individualized posterior models for Bayesian forecasting,
- or both approaches for complete external evaluation.

This function represents the final simulation step in the external evaluation workflow following `run_MAP_estimations()` and, when posterior predictions are required, `update_map_models()`.

Reproducibility of stochastic a priori simulations can be controlled using the `seed` argument.

Value

A named list containing:

simulation_results List of simulated concentration-time profiles for each individual and evaluation occasion.

ttoocc Treatment/event datasets grouped by occasion and used as simulation inputs.

eval_type Character string indicating the evaluation strategy inherited from `run_MAP_estimations()`.

events_tto Event datasets used for each simulation.

assessment Character string indicating the selected simulation strategy.

See Also

run_MAP_estimations(), update_map_models()

Examples

```
data("exeval_models", package = "exeval")
data("tacrolimus_pk1_kidney", package = "exeval")

dd <- tacrolimus_pk1_kidney |> subset(ID < 6)

fit <- run_MAP_estimations(
  model = exeval_models$Model_code[[2]],
  model_name = "TAC_Zuo2013",
  data = dd,
  evaluation_type = "sequential_updating"
)

post <- update_map_models(
  map_results = fit,
  evaluation_type = "sequential_updating"
)

sim <- run_pk_simulations(
  individual_model = post,
  map_results = fit,
  assessment = "Complete",
  seed = 123
)
```

---

tacrolimus_pk1_kidney *Tacrolimus pharmacokinetic data in kidney transplant patients*

Description

Pharmacokinetic and clinical data from adult kidney transplant recipients treated with tacrolimus, used for population pharmacokinetic model development, external evaluation, and methodological package examples.

Usage

```
tacrolimus_pk1_kidney
```

Format

A data.frame with 739 rows and 30 variables:

ID Patient identifier

OCC Number of the occasion

DD Tacrolimus daily dose (mg)

AMT Dose amount (mg)

TIME Sequential time (hours)

POD Post-operative days

DV Observed tacrolimus concentration (ng/mL)

EVID Event identifier

CMT Compartment identifier

MDV Missing dependent variable flag

II Dosing interval (hours)

SS Steady-state indicator

Creatine Creatinine (mg/dL)

SCR Serum creatinine ($\mu mol/L$)

eGFR Estimated glomerular filtration rate (mL/min/1.73 m^2)

ClCrea Creatinine clearance (Cockcroft-Gault, mL/min)

AGE Age (years)

SEX Sex

WT Body weight (kg)

HCT Hematocrit

CYP3A5 CYP3A5 polymorphism

EXPRESSION CYP3A5 expresser status

PDN_DOSE Prednisone dose (mg)

PDNXWT Prednisone dose normalized by body weight (mg/kg)

Height Height (cm)

Height..m. Height (m)

BSA Body surface area (m^2)

BMIcalc Body mass index (kg/m^2)

LBW Lean body weight (kg)

DMELITU Diabetes mellitus indicator

Details

This dataset corresponds to a Uruguayan kidney transplant cohort.

Source

De-identified clinical dataset adapted for methodological research and package examples.

References

Umpierrez M, et al. (2025). *Accelerating Tacrolimus Model-Informed Precision Dosing in Kidney Transplant Recipients: Model Evaluation and Refinement Strategies.*

`update_map_models` *Update MAP estimation objects with posterior individual parameters*

Description

Converts MAP estimation results obtained with `run_MAP_estimations()` into individualized posterior models using **mapbayr**. Depending on the selected evaluation strategy, posterior information is propagated across occasions to support posterior predictive simulations.

Usage

```
update_map_models(
  map_results,
  evaluation_type = c("sequential_updating", "stepwise_updating",
    "sequential_reference_updating", "backward_reference_updating")
)
```

Arguments

`map_results`	Named list returned by `run_MAP_estimations()`. Must contain at least the elements `map_estimations` and `eval_type`.
`evaluation_type`	Character string specifying the evaluation strategy. Must match the strategy used when generating `map_results`. Available options are: • `"sequential_updating"` • `"stepwise_updating"` • `"sequential_reference_updating"` • `"backward_reference_updating"`

Details

This function applies `mapbayr::use_posterior()` to each MAP estimation object and returns a list of updated individual models that can be used for posterior predictive simulations.

The evaluation strategy must match the one originally used in `run_MAP_estimations()`.

This function applies `mapbayr::use_posterior()` to each MAP estimation object contained in `map_results`, generating individualized posterior models for subsequent simulation.

Posterior information is propagated across occasions according to the selected `evaluation_type`, which must match the strategy originally used in `run_MAP_estimations()`.

Posterior model objects are dynamically named following the pattern `a.posteriori_occX_Y`, where `X` and `Y` indicate the occasions linked by the posterior update.

The resulting objects are intended for use with `run_pk_simulations()`.

Value

A named list containing:

ind_model List of posterior individualized model objects created using `mapbayr::use_posterior()`.

eval_type Character string indicating the evaluation strategy used.

Examples

```
data("exeval_models", package = "exeval")
data("tacrolimus_pk1_kidney", package = "exeval")

dd <- tacrolimus_pk1_kidney |> subset(ID < 6)

fit <- run_MAP_estimations(
  model = exeval_models$Model_code[[2]],
  model_name = "TAC_Zuo2013",
  data = dd,
  evaluation_type = "sequential_updating"
)

post <- update_map_models(
  map_results = fit,
  evaluation_type = "sequential_updating"
)
```

Index