Package 'posologyr'

September 13, 2024

Title Individual Dose Optimization using Population Pharmacokinetics

Version 1.2.7

Description Optimize drug regimens through model-informed precision dosing, using individual pharmacokinetic (PK) and pharmacokinetic-pharmacodynamic (PK-PD) profiles. By integrating therapeutic drug monitoring (TDM) data with population models, 'posologyr' provides accurate posterior estimates and enables the calculation of personalized dosing regimens. The empirical Bayes estimates are computed following the method described by Kang et al. (2012) <doi:10.4196/kjpp.2012.16.2.97>.

License AGPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports rxode2, stats, mvtnorm, data.table

Suggests lotri, rmarkdown, testthat (>= 3.0.0), ggplot2, magrittr, tidyr

URL https://levenc.github.io/posologyr/, https://github.com/levenc/posologyr

BugReports https://github.com/levenc/posologyr/issues

Config/testthat/edition 3

NeedsCompilation no

Author Cyril Leven [aut, cre, cph] (<https://orcid.org/0000-0002-0697-4370>), Matthew Fidler [ctb] (<https://orcid.org/0000-0001-8538-6691>), Emmanuelle Comets [ctb], Audrey Lavenu [ctb], Marc Lavielle [ctb]

Maintainer Cyril Leven <cyril.leven@chu-brest.fr>

Repository CRAN

Date/Publication 2024-09-13 07:10:02 UTC

Contents

`poso_dose_auc` *Estimate the dose needed to reach a target area under the concentration-time curve (AUC)*

Description

estimates the dose needed to reach a target area under the concentration-time curve (AUC) given a population pharmacokinetic model, a set of individual parameters, and a target AUC.

Usage

```
poso_dose_auc(
  dat = NULL,
  prior_model = NULL,
  tdm = FALSE,
  time_auc,
  time_dose = NULL,
  cmt_dose = 1,
  target_auc,
  estim_method = "map",
  nocb = FALSE,
  p = NULL,
  greater_than = TRUE,
  starting_time = 0,
  interdose_interval = NULL,
  add_dose = NULL,
  duration = 0,
  starting_dose = 100,
  indiv_param = NULL
)
```

Arguments

dat	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
prior_model	A posologyr prior population pharmacokinetics model, a list of six objects.
tdm	A boolean. If TRUE: estimates the optimal dose for a selected target auc over a selected duration following the events from dat, and using Maximum A Posteriori estimation. Setting tdm to TRUE causes the following to occur: • the time_dose argument is required and is used as the starting point for the AUC calculation instead of starting_time; • the arguments estim_method, p, greater_than, interdose_interval, add_dose, indiv_param and starting_time are ignored.
time_auc	Numeric. A duration. The target AUC is computed from starting_time to starting_time + time_auc. When tdm is set to TRUE the target AUC is computed from time_dose to time_dose + time_auc instead.
time_dose	Numeric. Time when the dose is to be given. Only used and mandatory, when tdm is set to TRUE.
cmt_dose	Character or numeric. The compartment in which the dose is to be administered. Must match one of the compartments in the prior model. Defaults to 1.
target_auc	Numeric. The target AUC.
estim_method	A character string. An estimation method to be used for the individual parameters. The default method "map" is the Maximum A Posteriori estimation, the method "prior" simulates from the prior population model, and "sir" uses the Sequential Importance Resampling algorithm to estimate the a posteriori distribution of the individual parameters. This argument is ignored if indiv_param is provided, or if tdm is set to TRUE.
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.
p	Numeric. The proportion of the distribution of AUC to consider for the optimization. Mandatory for estim_method=sir. This argument is ignored if tdm is set to TRUE.
greater_than	A boolean. If TRUE: targets a dose leading to a proportion p of the AUCs to be greater than target_auc. Respectively, lower if FALSE. This argument is ignored if tdm is set to TRUE.
starting_time	Numeric. First point in time of the AUC, for multiple dose regimen. The default is zero. This argument is ignored if tdm is set to TRUE, and time_dose is used as a starting point instead.
interdose_interval	Numeric. Time for the interdose interval for multiple dose regimen. Must be provided when add_dose is used. This argument is ignored if tdm is set to TRUE.
add_dose	Numeric. Additional doses administered at inter-dose interval after the first dose. Optional. This argument is ignored if tdm is set to TRUE.
duration	Numeric. Duration of infusion, for zero-order administrations.

`starting_dose`	Numeric. Starting dose for the optimization algorithm.
`indiv_param`	Optional. A set of individual parameters : THETA, estimates of ETA, and covariates. This argument is ignored if `tdm` is set to `TRUE`.

Value

A list containing the following components:

dose Numeric. An optimal dose for the selected target AUC.

type_of_estimate Character string. The type of estimate of the individual parameters. Either a point estimate, or a distribution.

auc_estimate A vector of numeric estimates of the AUC. Either a single value (for a point estimate of ETA), or a distribution.

indiv_param A `data.frame`. The set of individual parameters used for the determination of the optimal dose : THETA, estimates of ETA, and covariates

Examples

```
rxode2::setRxThreads(2L) # limit the number of threads

# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
```

```
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the optimal dose to reach an AUC(0-12h) of 45 h.mg/l
poso_dose_auc(dat=df_patient01,prior_model=mod_run001,
time_auc=12,target_auc=45)
```

---

poso_dose_conc *Estimate the optimal dose to achieve a target concentration at any given time*

Description

Estimates the optimal dose to achieve a target concentration at any given time given a population pharmacokinetic model, a set of individual parameters, a selected point in time, and a target concentration.

Usage

```
poso_dose_conc(
  dat = NULL,
  prior_model = NULL,
  tdm = FALSE,
  time_c,
  time_dose = NULL,
  target_conc,
  cmt_dose = 1,
  endpoint = "Cc",
  estim_method = "map",
  nocb = FALSE,
  p = NULL,
  greater_than = TRUE,
  starting_dose = 100,
  interdose_interval = NULL,
  add_dose = NULL,
  duration = 0,
  indiv_param = NULL
)
```

Arguments

dat	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
prior_model	A posologyr prior population pharmacokinetics model, a list of six objects.

tdm	A boolean. If TRUE: estimates the optimal dose for a selected target concentration at a selected point in time following the events from dat, and using Maximum A Posteriori estimation. Setting tdm to TRUE causes the following to occur: • the arguments estim_method, p, greater_than, interdose_interval, add_dose, indiv_param and starting_time are ignored.
time_c	Numeric. Point in time for which the dose is to be optimized.
time_dose	Numeric. Time when the dose is to be given.
target_conc	Numeric. Target concentration.
cmt_dose	Character or numeric. The compartment in which the dose is to be administered. Must match one of the compartments in the prior model. Defaults to 1.
endpoint	Character. The endpoint of the prior model to be optimised for. The default is "Cc", which is the central concentration.
estim_method	A character string. An estimation method to be used for the individual parameters. The default method "map" is the Maximum A Posteriori estimation, the method "prior" simulates from the prior population model, and "sir" uses the Sequential Importance Resampling algorithm to estimate the a posteriori distribution of the individual parameters. This argument is ignored if indiv_param is provided or if tdm is set to TRUE.
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.
p	Numeric. The proportion of the distribution of concentrations to consider for the optimization. Mandatory for estim_method=sir. This argument is ignored if tdm is set to TRUE.
greater_than	A boolean. If TRUE: targets a dose leading to a proportion p of the concentrations to be greater than target_conc. Respectively, lower if FALSE. This argument is ignored if tdm is set to TRUE.
starting_dose	Numeric. Starting dose for the optimization algorithm.
interdose_interval	Numeric. Time for the interdose interval for multiple dose regimen. Must be provided when add_dose is used. This argument is ignored if tdm is set to TRUE.
add_dose	Numeric. Additional doses administered at inter-dose interval after the first dose. Optional. This argument is ignored if tdm is set to TRUE.
duration	Numeric. Duration of infusion, for zero-order administrations.
indiv_param	Optional. A set of individual parameters : THETA, estimates of ETA, and covariates. This argument is ignored if tdm is set to TRUE.

Value

A list containing the following components:

dose Numeric. An optimal dose for the selected target concentration.

type_of_estimate Character string. The type of estimate of the individual parameters. Either a point estimate, or a distribution.

conc_estimate A vector of numeric estimates of the conc. Either a single value (for a point estimate of ETA), or a distribution.

indiv_param A `data.frame`. The set of individual parameters used for the determination of the optimal dose : THETA, estimates of ETA, and covariates

Examples

```
rxode2::setRxThreads(2L) # limit the number of threads

# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                           TIME=c(0.0,1.0,14.0),
                           DV=c(NA,25.0,5.5),
                           AMT=c(2000,0,0),
                           EVID=c(1,0,0),
                           DUR=c(0.5,NA,NA))
# estimate the optimal dose to reach a concentration of 80 mg/l
# one hour after starting the 30-minutes infusion
poso_dose_conc(dat=df_patient01,prior_model=mod_run001,
time_c=1,duration=0.5,target_conc=80)
```

poso_estim_map *Estimate the Maximum A Posteriori individual parameters*

Description

Estimates the Maximum A Posteriori (MAP) individual parameters, also known as Empirical Bayes Estimates (EBE).

Usage

```
poso_estim_map(
  dat = NULL,
  prior_model = NULL,
  return_model = TRUE,
  return_ofv = FALSE,
  nocb = FALSE
)
```

Arguments

dat	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
prior_model	A posologyr prior population pharmacokinetics model, a list of six objects.
return_model	A boolean. Returns a rxode2 model using the estimated ETAs if set to TRUE.
return_ofv	A boolean. Returns a the Objective Function Value (OFV) if set to TRUE.
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.

Value

A named list consisting of one or more of the following elements depending on the input parameters of the function: $eta a named vector of the MAP estimates of the individual values of ETA, $model an rxode2 model using the estimated ETAs, $event the data.table used to solve the returned rxode2 model.

Examples

```
rxode2::setRxThreads(1) # limit the number of threads

# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
```

```
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the Maximum A Posteriori individual parameters
poso_estim_map(dat=df_patient01,prior_model=mod_run001)
```

---

poso_estim_mcmc *Estimate the posterior distribution of individual parameters by MCMC*

Description

Estimates the posterior distribution of individual parameters by Markov Chain Monte Carlo (using a Metropolis-Hastings algorithm)

Usage

```
poso_estim_mcmc(
  dat = NULL,
  prior_model = NULL,
  return_model = TRUE,
  burn_in = 50,
  n_iter = 1000,
  n_chains = 4,
```

```
  nocb = FALSE,
 control = list(n_kernel = c(2, 2, 2), stepsize_rw = 0.4, proba_mcmc = 0.3, nb_max = 3)
)
```

Arguments

dat	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
prior_model	A posologyr prior population pharmacokinetics model, a list of six objects.
return_model	A boolean. Returns a rxode2 model using the estimated ETAs if set to TRUE.
burn_in	Number of burn-in iterations for the Metropolis-Hastings algorithm.
n_iter	Total number of iterations (following the burn-in iterations) for each Markov chain of the Metropolis-Hastings algorithm.
n_chains	Number of Markov chains
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.
control	A list of parameters controlling the Metropolis-Hastings algorithm.

Value

If return_model is set to FALSE, a list of one element: a dataframe $eta of ETAs from the posterior distribution, estimated by Markov Chain Monte Carlo. If return_model is set to TRUE, a list of the dataframe of the posterior distribution of ETA, and a rxode2 model using the estimated distributions of ETAs.

Author(s)

Emmanuelle Comets, Audrey Lavenu, Marc Lavielle, Cyril Leven

References

Comets E, Lavenu A, Lavielle M. Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software 80, 3 (2017), 1-41.

Examples

```
# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
```

```
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the posterior distribution of population parameters
poso_estim_mcmc(dat=df_patient01,prior_model=mod_run001,
n_iter=50,n_chains=2)
```

`poso_estim_sir` *Estimate the posterior distribution of individual parameters by SIR*

Description

Estimates the posterior distribution of individual parameters by Sequential Importance Resampling (SIR)

Usage

```
poso_estim_sir(
  dat = NULL,
  prior_model = NULL,
  n_sample = 10000,
  n_resample = 1000,
  return_model = TRUE,
  nocb = FALSE
)
```

Arguments

dat	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
prior_model	A posologyr prior population pharmacokinetics model, a list of six objects.
n_sample	Number of samples from the S-step
n_resample	Number of samples from the R-step
return_model	A boolean. Returns a rxode2 model using the estimated ETAs if set to TRUE.
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.

Value

If return_model is set to FALSE, a list of one element: a dataframe $eta of ETAs from the posterior distribution, estimated by Sequential Importance Resampling. If return_model is set to TRUE, a list of the dataframe of the posterior distribution of ETA, and a rxode2 model using the estimated distributions of ETAs.

Examples

```
# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
```

```
# infusion
df_patient01 <- data.frame(ID=1,
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the posterior distribution of population parameters
poso_estim_sir(dat=df_patient01,prior_model=mod_run001,
n_sample=1e3,n_resample=1e2)
```

`poso_inter_cmin` *Estimate the optimal dosing interval to consistently achieve a target trough concentration (Cmin)*

Description

Estimates the optimal dosing interval to consistently achieve a target Cmin, given a dose, a population pharmacokinetic model, a set of individual parameters, and a target concentration.

Usage

```
poso_inter_cmin(
  dat = NULL,
  prior_model = NULL,
  dose,
  target_cmin,
  cmt_dose = 1,
  endpoint = "Cc",
  estim_method = "map",
  nocb = FALSE,
  p = NULL,
  greater_than = TRUE,
  starting_interval = 12,
  add_dose = 10,
  duration = 0,
  indiv_param = NULL
)
```

Arguments

`dat`	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
`prior_model`	A `posologyr` prior population pharmacokinetics model, a list of six objects.
`dose`	Numeric. The dose given.
`target_cmin`	Numeric. Target trough concentration (Cmin).

cmt_dose	Character or numeric. The compartment in which the dose is to be administered. Must match one of the compartments in the prior model. Defaults to 1.
endpoint	Character. The endpoint of the prior model to be optimised for. The default is "Cc", which is the central concentration.
estim_method	A character string. An estimation method to be used for the individual parameters. The default method "map" is the Maximum A Posteriori estimation, the method "prior" simulates from the prior population model, and "sir" uses the Sequential Importance Resampling algorithm to estimate the a posteriori distribution of the individual parameters. This argument is ignored if indiv_param is provided.
nocb	A boolean. for time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If FALSE, the last observation carried forward (locf) style will be used. Defaults to FALSE.
p	Numeric. The proportion of the distribution of concentrations to consider for the optimization. Mandatory for estim_method=sir.
greater_than	A boolean. If TRUE: targets a dose leading to a proportion p of the concentrations to be greater than target_conc. Respectively, lower if FALSE.
starting_interval	Numeric. Starting inter-dose interval for the optimization algorithm.
add_dose	Numeric. Additional doses administered at inter-dose interval after the first dose.
duration	Numeric. Duration of infusion, for zero-order administrations.
indiv_param	Optional. A set of individual parameters : THETA, estimates of ETA, and covariates.

Value

A list containing the following components:

interval Numeric. An inter-dose interval to reach the target trough concentration before each dosing of a multiple dose regimen.

type_of_estimate Character string. The type of estimate of the individual parameters. Either a point estimate, or a distribution.

conc_estimate A vector of numeric estimates of the conc. Either a single value (for a point estimate of ETA), or a distribution.

indiv_param A data.frame. The set of individual parameters used for the determination of the optimal dose : THETA, estimates of ETA, and covariates

Examples

```
rxode2::setRxThreads(2L) # limit the number of threads

# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
```

```
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                           TIME=c(0.0,1.0,14.0),
                           DV=c(NA,25.0,5.5),
                           AMT=c(2000,0,0),
                           EVID=c(1,0,0),
                           DUR=c(0.5,NA,NA))
# estimate the optimal interval to reach a cmin of of 2.5 mg/l
# before each administration
poso_inter_cmin(dat=df_patient01,prior_model=mod_run001,
dose=1500,duration=0.5,target_cmin=2.5)
```

`poso_replace_et` *Update a model with events from a new rxode2 event table*

Description

Update a model with events from a new rxode2 event table, while accounting for and interpolating any covariates or inter-occasion variability.

Usage

```
poso_replace_et(
  target_model = NULL,
```

```
  prior_model = NULL,
  event_table = NULL,
  interpolation = "locf"
)
```

Arguments

target_model	Solved rxode2 object. A model generated by one of posologyr's estimation functions.
prior_model	A posologyr prior population model.
event_table	An rxode2 event table.
interpolation	Character string. Specifies the interpolation method to be used for covariates. Choices are "locf" for last observation carried forward, "nocb" for next observation carried backward, "midpoint", or "linear".

Value

A solved rxode2 object, updated with the event table provided.

Examples

```
# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
```

```
df_patient01 <- data.frame(ID=1,
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the prior distribution of population parameters
pop_model <- poso_simu_pop(dat=df_patient01,prior_model=mod_run001,n_simul=100)
# create a new rxode2 event table from the initial dataset
new_et <- rxode2::as.et(df_patient01)
new_et$add_sampling(seq(14,15,by=0.1))
# update the model with the new event table
poso_replace_et(pop_model$model,mod_run001,event_table=new_et)
```

`poso_simu_pop` *Estimate the prior distribution of population parameters*

Description

Estimates the prior distribution of population parameters by Monte Carlo simulations

Usage

```
poso_simu_pop(
  dat = NULL,
  prior_model = NULL,
  n_simul = 1000,
  return_model = TRUE
)
```

Arguments

`dat`	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
`prior_model`	A `posologyr` prior population pharmacokinetics model, a list of six objects.
`n_simul`	An integer, the number of simulations to be run. For `n_simul =0`, all ETAs are set to 0.
`return_model`	A boolean. Returns a rxode2 model using the simulated ETAs if set to `TRUE`.

Value

If `return_model` is set to `FALSE`, a list of one element: a dataframe `$eta` of the individual values of ETA. If `return_model` is set to `TRUE`, a list of the dataframe of the individual values of ETA, and a rxode2 model using the simulated ETAs.

Examples

```
# model
mod_run001 <- function() {
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                        TIME=c(0.0,1.0,14.0),
                        DV=c(NA,25.0,5.5),
                        AMT=c(2000,0,0),
                        EVID=c(1,0,0),
                        DUR=c(0.5,NA,NA))
# estimate the prior distribution of population parameters
poso_simu_pop(dat=df_patient01,prior_model=mod_run001,n_simul=100)
```

---

poso_time_cmin *Estimate the time required to reach a target trough concentration (Cmin)*

Description

Estimates the time required to reach a target trough concentration (Cmin) given a population pharmacokinetic model, a set of individual parameters, a dose, and a target Cmin.

Usage

```
poso_time_cmin(
  dat = NULL,
  prior_model = NULL,
  tdm = FALSE,
  target_cmin,
  dose = NULL,
  cmt_dose = 1,
  endpoint = "Cc",
  estim_method = "map",
  nocb = FALSE,
  p = NULL,
  greater_than = TRUE,
  from = 0.2,
  last_time = 72,
  add_dose = NULL,
  interdose_interval = NULL,
  duration = 0,
  indiv_param = NULL
)
```

Arguments

`dat`	Dataframe. An individual subject dataset following the structure of NONMEM/rxode2 event records.
`prior_model`	A `posologyr` prior population pharmacokinetics model, a list of six objects.
`tdm`	A boolean. If `TRUE`: computes the predicted time to reach the target trough concentration (Cmin) following the last event from `dat`, and using Maximum A Posteriori estimation. Setting `tdm` to `TRUE` causes the following to occur: • the simulation starts at the time of the last recorded dose (from the TDM data) plus `from`; • the simulation stops at the time of the last recorded dose (from the TDM data) plus `last_time`; • the arguments `dose`, `duration`, `estim_method`, `p`, `greater_than`, `interdose_interval`, `add_dose`, `indiv_param` and `starting_time` are ignored.
`target_cmin`	Numeric. Target trough concentration (Cmin).
`dose`	Numeric. Dose administered. This argument is ignored if `tdm` is set to `TRUE`.
`cmt_dose`	Character or numeric. The compartment in which the dose is to be administered. Must match one of the compartments in the prior model. Defaults to 1.
`endpoint`	Character. The endpoint of the prior model to be optimised for. The default is "Cc", which is the central concentration.
`estim_method`	A character string. An estimation method to be used for the individual parameters. The default method "map" is the Maximum A Posteriori estimation, the method "prior" simulates from the prior population model, and "sir" uses the

Sequential Importance Resampling algorithm to estimate the a posteriori distribution of the individual parameters. This argument is ignored if `indiv_param` is provided, or if `tdm` is set to `TRUE`.

`nocb` A boolean. For time-varying covariates: the next observation carried backward (nocb) interpolation style, similar to NONMEM. If `FALSE`, the last observation carried forward (locf) style will be used. Defaults to `FALSE`.

`p` Numeric. The proportion of the distribution of Cmin to consider for the estimation. Mandatory for `estim_method=sir`. This argument is ignored if `tdm` is set to `TRUE`.

`greater_than` A boolean. If `TRUE`: targets a time leading to a proportion `p` of the cmins to be greater than `target_cmin`. Respectively, lower if `FALSE`. This argument is ignored if `tdm` is set to `TRUE`.

`from` Numeric. Starting time for the simulation of the individual time-concentration profile. The default value is 0.2. When `tdm` is set to `TRUE` the simulation starts at the time of the last recorded dose plus `from`.

`last_time` Numeric. Ending time for the simulation of the individual time-concentration profile. The default value is 72. When `tdm` is set to `TRUE` the simulation stops at the time of the last recorded dose plus `last_time`.

`add_dose` Numeric. Additional doses administered at inter-dose interval after the first dose. Optional. This argument is ignored if `tdm` is set to `TRUE`.

`interdose_interval`
Numeric. Time for the inter-dose interval for multiple dose regimen. Must be provided when add_dose is used. This argument is ignored if `tdm` is set to `TRUE`.

`duration` Numeric. Duration of infusion, for zero-order administrations. This argument is ignored if `tdm` is set to `TRUE`.

`indiv_param` Optional. A set of individual parameters : THETA, estimates of ETA, and covariates.

Value

A list containing the following components:

time Numeric. Time needed to reach the selected Cmin.

type_of_estimate Character string. The type of estimate of the individual parameters. Either a point estimate, or a distribution.

cmin_estimate A vector of numeric estimates of the Cmin. Either a single value (for a point estimate of ETA), or a distribution.

indiv_param A `data.frame`. The set of individual parameters used for the determination of the time needed to reach a selected Cmin: THETA, estimates of ETA, and covariates

Examples

```
rxode2::setRxThreads(2L) # limit the number of threads

# model
mod_run001 <- function() {
```

```
  ini({
    THETA_Cl <- 4.0
    THETA_Vc <- 70.0
    THETA_Ka <- 1.0
    ETA_Cl ~ 0.2
    ETA_Vc ~ 0.2
    ETA_Ka ~ 0.2
    prop.sd <- sqrt(0.05)
  })
  model({
    TVCl <- THETA_Cl
    TVVc <- THETA_Vc
    TVKa <- THETA_Ka

    Cl <- TVCl*exp(ETA_Cl)
    Vc <- TVVc*exp(ETA_Vc)
    Ka <- TVKa*exp(ETA_Ka)

    K20 <- Cl/Vc
    Cc <- centr/Vc

    d/dt(depot) = -Ka*depot
    d/dt(centr) = Ka*depot - K20*centr
    Cc ~ prop(prop.sd)
  })
}
# df_patient01: event table for Patient01, following a 30 minutes intravenous
# infusion
df_patient01 <- data.frame(ID=1,
                           TIME=c(0.0,1.0,14.0),
                           DV=c(NA,25.0,5.5),
                           AMT=c(2000,0,0),
                           EVID=c(1,0,0),
                           DUR=c(0.5,NA,NA))
# predict the time needed to reach a concentration of 2.5 mg/l
# after the administration of a 2500 mg dose over a 30 minutes
# infusion
poso_time_cmin(dat=df_patient01,prior_model=mod_run001,
dose=2500,duration=0.5,from=0.5,target_cmin=2.5)
```

Index