Package 'tci'

February 26, 2021

Title Target Controlled Infusion (TCI)

Version 0.1.0

Description Implementation of target-controlled infusion algorithms for compartmental pharmacokinetic and pharmacokinetic-pharmacodynamic models. Jacobs (1990) <doi:10.1109/10.43622>; Marsh et al. (1991) <doi:10.1093/bja/67.1.41>; Shafer and Gregg (1993) <doi:10.100
199805000-00006>; Abuhelwa, Foster, and Upton (2015) <doi:10.1016/j.vascn.2015.03.004>; Eleveld et al. (2018) <doi:10.1016/j.bja.2018.01.018>.

Depends R (>= 3.6.0)

License GPL-2

Encoding UTF-8

LazyData true

URL https://github.com/jarretrt/tci

BugReports https://github.com/jarretrt/tci/issues

Imports ggplot2, stats, utils, graphics, truncnorm, mvtnorm, gridExtra, reshape

Suggests testthat (>= 2.1.0), knitr, rmarkdown, mrgsolve

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Ryan Jarrett [aut, cre]

Maintainer Ryan Jarrett <ryan.t.jarrett@vanderbilt.edu>

Repository CRAN

Date/Publication 2021-02-26 09:30:02 UTC

R topics documented:

`apply_poppk` *Apply a population PK model to a data frame*

Description

Function to apply saved population PK or PK-PD models to a data frame of patient values.

Usage

```
apply_poppk(patient_df, mod = c("marsh", "schnider", "eleveld"), ...)
```

Arguments

patient_df	Dataframe with patient covariate values. Must have names used by model "mod"
mod	Population PK model to apply to rows of patient_df
...	Arguments passed on to population PK model.

Value

The function applies either the Marsh (1991), Schnider et al. (1998), or Eleveld et al (2018) PK models for propofol to a dataframe with patient covariate values and returns PK (or PK-PD, in the case of the Eleveld model) parameter values. The function returns the original dataframe merged with parameter values for each individual (in rows).

---

apply_targetfn *#' Sigmoid target function #' #' @param lpars Logged parameter values #' @param tms Times to evaluate sigmoid function #' @param bis0 BIS value with no drug administered #' @param ... Arguments passed on to 'restrict_sigmoid' function #' @return Returns numeric vector of BIS values corresponding to an Emax sigmoidal #' target function with parameter bis0. #' @export sigmoid_targetfn <- function(lpars, tms, bis0 = 93, ...) emax(tms, restrict_sigmoid(t50 = exp(lpars), BIS0 = bis0, ...)) Apply target function to a PK-PD model*

Description

Function to apply any specified target function to a PK-PD model and TCI algorithm. 'targetfn' should be a function with parameters 'lp' as the first argument and times 'tm' as the second.

Usage

```
apply_targetfn(
  lp,
  tm,
  targetfn,
  prior_pk,
  prior_pd,
  pkmod = pkmod3cptm,
  pdmod = emax_eleveld,
  pdinv = inv_emax_eleveld,
  ...
)
```

Arguments

lp	Logged parameter values
tm	Time values to evaluate
targetfn	Target function

prior_pk	Prior PK point estimates
prior_pd	Prior PD point estimates
pkmod	PK model to evaluate
pdmod	PD model to evaluate
pdinv	Inverse PD model
...	Additional arguments passed on to tci_pd

Value

Return a set of infusions designed to reach a target function specified in argument 'targetfn'.

---

assign_pars *Set default PK parameter values Set default PK parameter values for a pkmod object.*

Description

Set default PK parameter values Set default PK parameter values for a pkmod object.

Usage

```
assign_pars(pkmod, pars)
```

Arguments

pkmod	pkmod object
pars	PK parameters to assign as default values of pkmod

Value

Returns a "pkmod" object with default parameters set to "pars'.

---

bayes_control *Bayesian closed-loop control*

Description

Function to provide Bayesian closed-loop control.

Usage

```
bayes_control(
  targets,
  updates,
  prior,
  true_pars,
  pkmod = pkmod3cptm,
  pdmod = emax_eleveld,
  pdinv = inv_emax_eleveld,
  init0 = NULL,
  init_p = NULL,
  obs_tms = NULL,
  dt_obs = 1/6,
  sim_starttm = 0,
  tci_alg = "effect",
  print_progress = FALSE
)
```

Arguments

targets	Data frame with columns ("time","target")
updates	Data frame of times at which closed-loop updates should be conducted and optional variable with logical values named 'full_data' indicating if full updates should be used. Defaults to partial.
prior	List with elements "mu" and "sig" specifying the prior mean and covariance matrices for the logged parameter values.
true_pars	Vector of true patient PK-PD parameters.
pkmod	PK model
pdmod	PD model
pdinv	Inverse PD model
init0	True initial concentrations
init_p	Predicted initial concentrations
obs_tms	Times at which observations are collected. If null, observations will be made at fixed intervals specified by 'dtm'.
dt_obs	Interval between measurements.
sim_starttm	Start time of simulation
tci_alg	TCI algorithm used. Defaults to effect-site targeting.
print_progress	Logical. Should current update times be printed to the console.

Value

Returns a list with class "bayessim" containing results from the Bayesian closed-loop simulation. A "plot.bayessim" method exists for visualizing results.

cl_targets *Closed-loop targets*

Description

Format data frame of closed-loop targets.

Usage

```
cl_targets(time, target)
```

Arguments

time	Times at which target values are set
target	Response target values

Value

Returns a data frame with columns "time" and "target" as are required for use in the "bayes_control" function.

cl_updates *Closed-loop updates*

Description

Set parameters for closed-loop updates.

Usage

```
cl_updates(time, full_data = TRUE, plot_progress = FALSE)
```

Arguments

time	Times at which PK or PK-PD parameters should be updated
full_data	Vector of logical values indicating if all data should be used at update or only data since last update. Once first "FALSE" value is observed, the prior variance-covariance matrix is overwritten. Consequently, any "TRUE" updates will only use data since the last "FALSE" update.
plot_progress	Vector of logical values. Should values be plotted at each update?

Value

Returns a data frame with columns "time", "full_data", and "plot_progress" that are used as simulation parameters in the "bayes_control" function.

`combine_sim` *Combine simulation outputs*

Description

Function to merge objects with class datasim from different infusion schedules infusion schedules can be passed directly in or as a list.

Usage

```
combine_sim(...)
```

Arguments

`...`	Set of datasim objects created from ‘gen_data‘ function.

Value

Return a list of merged data simulation objects, each with class ’datasim’. The resulting list also retains class ’datasim’. A "plot.datasim" method exists for visualizing results.

`create_intvl` *dosing schedule Create dosing schedule*

Description

Create a dosing schedule object with columns "infrt", "begin", "end" from vectors of infusions and infusion end times. The argument "inittm" is used to specify the starting time of the first infusion.

Usage

```
create_intvl(dose, inittm = 0)
```

Arguments

`dose`	Data frame with columns "time" and "infrt".
`inittm`	Starting time of initial infusion

Value

Returns a matrix of infusions with columns "infrt", "begin", and "end" indicating infusion rates and corresponding begin and end times, respectively.

`eleveld_pd` *Eleveld et al. pharmacodynamic data*

Description

Empirical Bayes (EB) estimates of PD parameters made by the Eleveld et al (2018) PK-PD model. EB estimates were calculated using the PK-PD datasets and NONMEM files provided by Eleveled et al. (2018). The original datasets were obtained through the Open TCI Initiative website (opentci.org) and based on contributions from a number of researchers who made their datasets publically available.

Usage

```
data(eleveld_pd)
```

Format

A data frame with 122 rows and 15 variables:

ID Patient ID

E50 EB estimate of effect-site concentration required to achieve 50 percent response

KE0 EB estimate of elimination rate from effect-site compartment

EMAX EB estimate of baseline bispectral index (BIS) with no drug administered

GAM EB estimate of Hill parameter when the effect-site concentration is less than E50

GAM1 EB estimate of Hill parameter when the effect-site concentration is greater than than E50

RESD EB estimate of residual error term

ALAG1 Estimated time lag in BIS measurements due to patient age (fixed-effects only)

AGE Patient's age (years)

WGT Patient's weight (kg)

HGT Patient's height (cm)

M1F2 Patient's sex: male = 1, female = 2

A1V2 Sampling site: arterial sampling = 1, venous sampling = 2

PMA Patient's post-menstrual age. Assumed to be age + 40 weeks if not provided

TECH Presence of concomitant anaesthetic techniques (Local anesthetic = 1, Opioids = 2)

References

Eleveld et al. (2018) British Journal of Anesthesia Vol. 120, 5:942-959 (BJA)

`eleveld_pk` *Eleveld et al. pharmacokinetic data*

Description

Empirical Bayes (EB) estimates of PK parameters for the Eleveld et al. (2018) PK-PD model. EB estimates were calculated using the PK-PD datasets and NONMEM files provided by Eleveled et al. (2018). The original datasets were obtained through the Open TCI Initiative website (opentci.org) and based on contributions from a number of researchers who made their datasets publically available.

Usage

```
data(eleveld_pk)
```

Format

A data frame with 1033 rows and 16 variables:

ID Patient ID

V1 EB estimate of first compartment volume

V2 EB estimate of second compartment volume

V3 EB estimate of third compartment volume

CL EB estimate of clearance for the first compartment

Q2 EB estimate of inter-compartmental clearance for second compartment

Q3 EB estimate of inter-compartmental clearance for third compartment

AGE Patient's age (years)

WGT Patient's weight (kg)

HGT Patient's height (cm)

M1F2 Patient's sex: male = 1, female = 2

PMA Patient's post-menstrual age. Assumed to be age + 40 weeks if not provided

TECH Presence of concomitant anaesthetic techniques (Local anesthetic = 1, Opioids = 2)

BMI Patient's BMI

FFM Patient's fat-free mass (FFM)

A1V2 Sampling site: arterial sampling = 1, venous sampling = 2

References

Eleveld et al. (2018) British Journal of Anesthesia Vol. 120, 5:942-959 (BJA)

`eleveld_poppk` *Eleveld population PK model*

Description

Function to generate PK-PD parameters for the Eleveld et al. (2018) PK-PD model. Function takes a data frame of patient covariate values with variable names "AGE","PMA","WGT","HGT","M1F2","TECH",and "A1V2" and returns PK parameter values. PK-PD values are generated at the point estimates for each patient (i.e. based on covariates alone) by default; however, random sets of parameter values can be drawn through Monte Carlo simulation based on the random effect/population variance terms described in Eleveld et al.

Usage

```
eleveld_poppk(df, PD = TRUE, rate = FALSE, rand = FALSE)
```

Arguments

`df`	Data frame with variable names "AGE","PMA","WGT","HGT","M1F2","TECH",and "A1V2"
`PD`	Logical. Should PD parameters be returned in addition to PK parameters. Defaults to TRUE.
`rate`	Logical. Should rate parameters be returned rather than clearance. Defaults to FALSE
`rand`	Logical. Should a vector of Monte Carlo samples be returned instead of point estimates at patient covariate values. Defaults to FALSE.

Value

Returns the original data frame with columns added corresponding to patient PK-PD parameter values.

Examples

```
dat <- data.frame(AGE  = c(20,40,65),
                  TBM  = c(50,70,90),
                  HGT  = c(150,170,200),
                  MALE = c(TRUE,FALSE,TRUE))

schnider_poppk(dat, rand = FALSE, rate = FALSE)
schnider_poppk(dat, rand = TRUE, rate = TRUE)
```

eleveld_vcov *Generate variance-covariance matrix for Eleveld PK-PD model*

Description

Generate the variance-covariance matrix for Eleveld PK-PD model for an observation via Monte Carlo sampling.

Usage

```
eleveld_vcov(
  dat,
  N = 1000,
  rates = TRUE,
  varnames = c("K10", "K12", "K21", "K13", "K31", "V1", "V2", "V3", "KE0", "CE50",
    "SIGMA")
)
```

Arguments

dat	Data frame of observed patient covariates
N	Number of Monte Carlo samples
rates	Logical. Should rate constants be calculated
varnames	Column names of variables used to calculate variance-covariance matrix

Value

Returns a variance-covariance matrix as a numeric matrix.

elvdlpars *Get logged parameters updated in Eleveld model*

Description

Extract the logged parameter values to be updated within the Eleveld model from a data frame of patient PK-PD values.

Usage

```
elvdlpars(x, pd = TRUE)
```

Arguments

x	Vector or data frame with Eleveld PK-PD model parameters
pd	Logical. Should PD parameters be returned in addition to PK parameters.

Value

Returns a numeric vector.

`emax` *Emax function*

Description

Emax function. c50 is the concentration eliciting a 50 identifying the slope of the Emax curve at c50, E0 is the response value with no drug present, Emx is the maximum effect size.

Usage

```
emax(ce, pars)
```

Arguments

`ce`	Vector of effect-site concentrations.
`pars`	Named vector of parameter values with names (c50,gamma,e0,emx).

Value

Returns a numeric vector of predictions from a 4-parameter Emax model.

Examples

```
pars_emax <- c(c50 = 1.5, gamma = 1.47, e0 = 100, emx = 100)
ce_seq <- seq(0,4,0.1)
plot(ce_seq, emax(ce_seq, pars_emax), type = "l",
xlab = "Effect-site concentrtion (ug/mL)", ylab = "BIS")
```

`emax_eleveld` *Emax function for Eleveld (2018) model.*

Description

The parameter gamma takes one of two values depending on whether ce <= c50.

Usage

```
emax_eleveld(ce, pars)
```

Arguments

ce	Vector of effect-site concentrations.
pars	A vector of parameter values of length 5 corresponding to parameters (c50,gamma,gamma2,E0,Emx). 'c50' describes the concentration required for 50 and 'gamma2' are the Hill parameter for concentrations before and after c50, respectively, 'e0' is the response value at a concentration of 0 and 'emx' is the maximum change in response.

Value

Returns a numeric vector of predictions from the Emax model used by Eleveld et al. (2018).

Examples

```
pars_emax_eleveld <- c(c50 = 1.5, gamma = 1.47, gamma2 = 1.89, e0 = 100, emx = 100)
ce_seq <- seq(0,4,0.1)
plot(ce_seq, emax_eleveld(ce_seq, pars_emax_eleveld), type = "l",
xlab = "Effect-site concentrtion (ug/mL)", ylab = "BIS")
```

---

gen_data *Function to simulate data from a specified PK or PK-PD model with a specified infusion schedule.*

Description

Function to simulate data from a specified PK or PK-PD model with a specified infusion schedule.

Usage

```
gen_data(
  inf,
  pkmod,
  pars_pk0,
  sigma_add = 0,
  sigma_mult = 0,
  log_err = FALSE,
  init = NULL,
  tms = NULL,
  pdmod = NULL,
  pars_pd0 = NULL,
  ecmpt = NULL,
  delay = 0,
  max_pdval = 100,
  min_pdval = 0
)
```

Arguments

inf	An infusion rate object outputted from either the 'create_intvl' function or the 'iterate_tci_grid' function
pkmod	PK model
pars_pk0	"True" parameter estimates used to simulate data observations.
sigma_add	Additive residual error standard deviation.
sigma_mult	Multiplicative residual error standard deviation.
log_err	Logical. Should the error be log-normally distributed?
init	Initial concentrations.
tms	Observation times. Defaults to beginning of each infusion if unspecified.
pdmod	PD model if applicable.
pars_pd0	PD model parameters if applicable.
ecmpt	Effect-site compartment number. Defaults to last compartment.
delay	Delay between generation and observation of measurements.
max_pdval	Maximum PD value.
min_pdval	Minimum PD value

Value

Returns a list of simulated responses with class 'datasim' corresponding to the infusion schedule provided. A "plot.datasim" method exists for visualizing results.

---

inv_emax *Inverse Emax function*

Description

Inverse Emax function to return effect-site concentrations required to reach target effect.

Usage

```
inv_emax(pdresp, pars)
```

Arguments

pdresp	PD response values
pars	Named vector of parameter values with names (c50,gamma,E0,Emx).

Value

Returns a numeric vector of effect-site concentrations corresponding to responses from a 4-parameter Emax model.

Examples

```
pars_emax <- c(c50 = 1.5, gamma = 4, e0 = 100, emx = 100)
ce_seq <- seq(0,4,0.1)
all.equal(inv_emax(emax(ce_seq, pars_emax), pars_emax), ce_seq)
```

`inv_emax_eleveld` *Inverse Emax function*

Description

Inverse of Emax function used by Eleveld et al. (2018) population PK model. The Emax function has five parameters: the

Usage

```
inv_emax_eleveld(pdresp, pars)
```

Arguments

`pdresp`	PD response values
`pars`	A vector of parameter values of length 5 corresponding to parameters (c50,gamma,gamma2,E0,Emx). 'c50' describes the concentration required for 50 and 'gamma2' are the Hill parameter for concentrations before and after c50, respectively, 'e0' is the response value at a concentration of 0 and 'emx' is the maximum change in response.

Value

Returns a numeric vector of effect-site concentrations calculated by inverting the Emax model used by Eleveld et al. (2018).

Examples

```
pars_emax_eleveld <- c(c50 = 1.5, gamma = 1.47, gamma2 = 1.89, e0 = 100, emx = 100)
ce_seq <- seq(0,4,0.1)
all.equal(inv_emax_eleveld(emax_eleveld(ce_seq, pars_emax_eleveld), pars_emax_eleveld), ce_seq)
```

log_likelihood *Evaluate log-likelihood*

Description

Function to evaluate the log likelihood given a set of logged parameter values and a set of observed BIS values. It is assumed that the full set of parameters are given by indices (pk_ix, pd_ix), of which a subset may be fixed (i.e. not updated, but still used to evaluate PK-PD functions).

Usage

```
log_likelihood(lpr, dat, pk_ix, pd_ix, fixed_ix = NULL, fixed_lpr = NULL)
```

Arguments

lpr	Set of logged PK-PD-error parameter values to be updated. The final value of lpr is assumed to be the residual error term.
dat	data frame with columns c("time","bis") corresponding to observed time and bis values
pk_ix	indices of (pars_pk,pars_pd) corresponding to PK function values
pd_ix	indices of (pars_pk,pars_pd) corresponding to PD function values
fixed_ix	indices of (pars_pk,pars_pd) corresponding to PD function values
fixed_lpr	values used by PD function that are not updated.

Value

Returns a numeric value representing the logged likelihood value associated with parameter values 'lpr' and observed data in matrix 'dat'. Observations are assumed to follow a truncated normal distribution with lower bound 0 if only a PK model is specified and bounded by 0 and 100 if a PD model is also provided.

log_posterior_neg *Function to evaluate the negative log posterior given a set of logged parameter values and observed BIS values.*

Description

Function to evaluate the negative log posterior given a set of logged parameter values and observed BIS values.

Usage

```
log_posterior_neg(lpr, dat, mu, sig, ...)
```

Arguments

lpr	logged PK-PD-error parameter values
dat	data frame with columns corresponding to observed time and PD response values.
mu	Mean of prior distribution.
sig	Variance-covariance matrix of prior distribution.
...	Arguments passed on to log-likelihood.

Value

Returns a numeric value representing the negative logged-posterior probability associated with a vector parameter values 'lpr' with observations specified in 'dat' and prior parameters 'mu' and 'sigma'. Observations are assumed to follow a (potentially truncated) normal distribution described in function 'log_likelihood'. Parameter values have a multivariate normal prior distribution.

---

log_prior *Calculate logged prior value*

Description

Function to return the prior probability for a set of parameters assuming a log-normal distribution.

Usage

```
log_prior(lpr, mu, sig)
```

Arguments

lpr	log parameter values to evaluate
mu	mean for model parameters and mean residual error
sig	variance covariance matrix for model parameters

Value

Returns a numeric value representing the logged prior value associated with parameter values 'lpr' for a multivariate normal distribution with mean vector 'mu' and variance-covariance matrix 'sig'.

marsh_poppk *Marsh population PK model.*

Description

Takes in a data frame with a column labeled "TBM" providing the total body mass of patients. The function returns the original data frame with additional columns describing estimated patient PK parameters based on the Marsh population PK model. The effect-site elimination parameter, KE0, is set to 1.2 in accordance with recommendations from Absalom et al., 2009 "Pharmacokinetic models for propofol- Defining and illuminating the devil in the detail"

Usage

```
marsh_poppk(df, rate = TRUE)
```

Arguments

df	data frame with column titled "TBM" giving patient total body mass in kg.
rate	Logical. Should elimination rate constants be returned instead of clearance parameters.

Details

——————————————————————————— Population PK and PK-PD functions ——————————————————————— ———————————————————————————

Value

Returns the original data frame entered into function with additional columns appended describing patient PK parameter values.

pal *Color palate for tci plotting functions*

Description

Color palate for tci plotting functions

Usage

```
pal
```

Format

An object of class `character` of length 7.

Value

Returns a named vector of color hex values.

`pkmod1cpt` *One compartment IV infusion with first-order elimination.*

Description

One compartment IV infusion with first-order elimination.

Usage

```
pkmod1cpt(tm, kR, pars, init = 0, inittm = 0)
```

Arguments

`tm`	Vector of times to evaluate the PK function at
`kR`	Infusion rate (e.g. ml/min).
`pars`	Named vector of parameters with names ('ke','v') or ('cl').
`init`	Initial concentration. Defaults to 0.
`inittm`	Time of initiation of infusion. Defaults to 0.

Value

Returns a vector of numeric values describing predicted concentrations for a one-compartment model with IV infusion.

Examples

```
pkmod1cpt(1,1,c(ke = 0.5, v = 1))
```

`pkmod2cpt` *Two compartment IV infusion with first-order elimination.*

Description

Two compartment IV infusion with first-order elimination.

Usage

```
pkmod2cpt(tm, kR, pars, init = c(0, 0), inittm = 0, k20 = 0)
```

Arguments

tm	Vector of times to evaluate the PK function at
kR	Infusion rate (e.g. ml/min).
pars	Named vector of parameters with names ('K10','K12','K21','V1','V2') or ('CL','Q','V1','V2').
init	Initial concentration. Defaults to 0 in both compartments.
inittm	Time of initiation of infusion. Defaults to 0.
k20	Elimination rate constant for second compartment. Defaults to 0.

Value

Returns a numeric matrix with predicted concentrations for a two-compartment model with IV infusion. Compartments are represented in rows while times are in columns.

Examples

```
pkmod2cpt(1,1,c(CL = 15, V1 = 10, Q = 10, V2 = 20))
```

---

pkmod3cpt *Three compartment IV infusion with first-order elimination.*

Description

Three compartment IV infusion with first-order elimination.

Usage

```
pkmod3cpt(tm, kR, pars, init = c(0, 0, 0), inittm = 0, k20 = 0, k30 = 0)
```

Arguments

tm	Vector of times to evaluate the PK function at
kR	Infusion rate (e.g. ml/min).
pars	Named vector of parameters with names ('K10','K12','K21','V1','V2') or ('CL','Q','V1','V2').
init	Initial concentration. Defaults to 0 in all compartments.
inittm	Time of initiation of infusion. Defaults to 0.
k20	Elimination rate constant for second compartment. Defaults to 0.
k30	Elimination rate constant for second compartment. Defaults to 0.

Value

Returns a numeric matrix with predicted concentrations for a three-compartment model with IV infusion. Compartments are represented in rows while times are in columns.

Examples

```
pkmod3cpt(1,1,c(CL = 15, Q2 = 10, Q3 = 5, V1 = 10, V2 = 20, V3 = 50))
```

pkmod3cptm *Solution to three-compartment IV model with effect-site*

Description

3 compartment IV infusion with first-order absorption between compartments and with an additional effect-site compartment. The analytical solutions implemented in this function are provided in "ADVAN-style analytical solutions for common pharmacokinetic models" by Abuhelwa et al. 2015.

Usage

```
pkmod3cptm(
  tm,
  kR,
  pars,
  init = c(0, 0, 0, 0),
  inittm = 0,
  returncpt = c("all", "cpt1", "cpt2", "cpt3", "cpt4")
)
```

Arguments

tm	Vector of times to evaluate the PK function at
kR	Infusion rate (e.g. ml/min).
pars	Named vector of parameters with names (k10,k12,k21,k13,k31,v1,v2,v3,ke0)
init	Initial concentration
inittm	Time of initiation of infusion
returncpt	Optionally specify a single compartment to return concentrations for. Defaults to returning all compartment concentrations.

Details

This function takes in arguments for each of the absorption and elimination rate constants of a three-compartment model as well as initial concentrations, c0. ke0 gives the rate of elimination from the effect-site compartment into the central compartment (i.e. k41). The rate of absorption into the effect-site compartment is set at 1/10,000 the value of ke0. The function returns a set of functions that calculate the concentration in each of the four compartments as a function of time.

Value

Returns a numeric matrix with predicted concentrations for a three-compartment model with a metabolite/effect-site compartment and IV infusion. Compartments are represented in rows while times are in columns.

Examples

```
pars_3cpt <- c(k10=1.5,k12=0.15,k21=0.09,k13=0.8,k31=0.8,v1=10,v2=15,v3=100,ke0=1)
pkmod3cptm(1,1,pars_3cpt)
```

`plot.pkmod` *Plot object with class 'pkmod'*

Description

Will show predicted concentrations in compartments associated with an infusion schedule.

User can provide a series of effect-site concentrations and a PD model or an infusion schedule with a PK-PD model.

Plot output returned by "bayes_control" function.

Usage

```
## S3 method for class 'pkmod'
plot(x, ..., inf, npts = 1000, title = NULL)

## S3 method for class 'pdmod'
plot(
  x,
  ...,
  pkmod,
  inf,
  pars_pd,
  pars_pk = NULL,
  npts = 1000,
  plot_pk = TRUE,
  title = NULL,
  ecmpt = NULL
)

## S3 method for class 'tciinf'
plot(x, ..., title = NULL, display = TRUE)

## S3 method for class 'datasim'
plot(x, ..., pars_prior = NULL, pars_post = NULL, pk_ix = NULL, pd_ix = NULL)

## S3 method for class 'bayessim'
plot(x, ...)
```

Arguments

`x`	Object returned from "bayes_control" function
`...`	...

inf	An infusion schedule object with columns "begin","end","infrt".
npts	Number of points used to evaluate predicted concentrations.
title	Title of plot.
pkmod	PK model
pars_pd	Parameters used by pdmod.
pars_pk	Parameters used by pkmod.
plot_pk	Logical. Should PK concentrations be plotted alongside the PD response. Defaults to TRUE.
ecmpt	Effect-site compartment number. Defaults to the last compartment concentration returned by pkmod.
display	Logical. Should plots be printed or returned as an arrangeGrob object?
pars_prior	Named vector of prior PK or PK-PD parameters
pars_post	Named vector of posterior PK or PK-PD parameters
pk_ix	Indicies of parameter vector(s) corresponding to PK parameters
pd_ix	Indicies of parameter vector(s) corresponding to PD parameters

Value

Returns a ggplot2 object displaying concentrations over time associated with object x with class 'pkmod'.

Returns a ggplot2 object displaying responses and optional concentrations over time associated with object x with class 'pkmod'.

Returns a gridExtra grob object displaying responses and concentrations over time associated with an infusion schedule with class 'tciinf'.

Returns a ggplot2 plot displaying simulated results from an object with class 'datasim'.

Returns a ggplot2 plot displaying results stored in a object with class 'bayessim'.

---

predict.pkmod *Predict concentrations from a pkmod object*

Description

predict method to apply pk model piecewise to infusion schedule

Usage

```
## S3 method for class 'pkmod'
predict(
  object,
  ...,
  inf,
  tms = NULL,
```

```
  dtm = 1/6,
  return_init = FALSE,
  remove_bounds = TRUE
)
```

Arguments

object	An object with class pkmod.
...	Arguments passed on to pkmod
inf	An infusion schedule object with columns "begin","end","infrt".
tms	Times to evaluate predictions at. Will default to a sequence spanning the infusions at intervals of dtm.
dtm	Interval used for prediction if argument tms is unspecified.
return_init	Logical indicating if concentrations at time 0 should be returned. Defaults to FALSE.
remove_bounds	Logical, indicating if concentrations calculated at changes in infusion rates should be returned if not included in prediction times. Defaults to TRUE, so that only concentrations at specified times are returned.

Value

Returns predicted concentrations associated with 'object' with class 'pkmod' and infusion schedule 'inf'.

---

restrict_sigmoid *Restrict target sigmoid values*

Description

Function to place restriction on gamma and E50 parameters of target sigmoid such that it passes through point (tfinal, BISfinal+eps)

Usage

```
restrict_sigmoid(t50, tfinal = 10, eps = 1, BIS0 = 100, BISfinal = 50 - eps)
```

Arguments

t50	parameter of Emax model
tfinal	end of the induction period
eps	distance between BISfinal and the target function at tfinal
BIS0	starting BIS value
BISfinal	asymptote of Emax model

Value

Returns a numeric vector of Emax sigmoidal parameters restricted to pass through point (tfinal, BISfinal+eps).

Examples

```
dose <- data.frame(time = c(0.5,4,4.5,10), infrt = c(100,0,100,0))
create_intvl(dose)
```

`schnider_poppk` *Schnider population PK model*

Description

Evaluate Schnider population PK model at patient covariate values. The function takes in a data frame with columns "AGE","TBM","HGT","MALE" corresponding to patient covariate values and returns the same data frame with additional columns added corresponding to patient PK parameter values described by the Schnider population PK model.

Usage

```
schnider_poppk(df, rate = FALSE, rand = FALSE)
```

Arguments

`df`	data frame with variable names "AGE","TBM","HGT","MALE"
`rate`	Logical. Should rate parameters be returned rather than clearance. Defaults to FALSE
`rand`	Logical. Should a vector of Monte Carlo samples be returned instead of point estimates at patient covariate values. Defaults to FALSE.

Value

Returns the original data frame entered into function with additional columns appended describing patient PK parameter values.

Examples

```
marsh_poppk(data.frame(TBM = c(50,70,90)))
```

seqby *Sequence including bounds*

Description

Create a sequence between two values at regular intervals and include bounds in output.

Usage

```
seqby(from, to, by)
```

Arguments

from	sequence starting value
to	sequence end value
by	increment of the sequence

Value

Returns a sequence of numeric or integer values.

Examples

```
tail_vec(1:8)
tail_vec(matrix(1:8,2,4))
```

tail_vec *Extract last element or column*

Description

Function to extract the last element from a vector or the last column from a matrix

Usage

```
tail_vec(x)
```

Arguments

x	Vector or matrix

Value

Returns either a single numeric value if 'x' is a vector or a vector of numeric values if 'x' is a matrix.

`tci` *Apply TCI algorithm*

Description

Function to iterate any arbitrary TCI algorithm to a series of points. By default, the function will update infusion rates at fixed intervals (e.g. every 10 seconds); however, users will have the option of waiting only calculating infusions after the prior target has been obtained.

Usage

```
tci(
  Ct,
  tms,
  pkmod,
  pars,
  init = NULL,
  tci_alg = c("effect", "plasma"),
  tci_custom = NULL,
  dtm = 1/6,
  ...
)
```

Arguments

`Ct`	Vector of target concentrations
`tms`	Times at which the TCI algorithm should try to achieve the target concentrations
`pkmod`	PK model
`pars`	PK model parameters
`init`	Initial concentrations for PK model
`tci_alg`	TCI algorithm. Options are provided for effect-site (default) or plasma targeting. Alternate algorithms can be specified through the 'tci_custom' argument.
`tci_custom`	Custom TCI algorithm. Algorithm should have arguments specifying target concentration, PK model, and duration of infusion to reach the target.
`dtm`	Time difference between infusion rate updates.
`...`	Arguments passed on to TCI algorithm.

Details

The user passes the 'iterate_tci' function a matrix of target concentrations and times at which the target is set. This is translated into a step function that defines the concentration target at all times.

Value

Returns a vector of numeric values.

tci_comb — *Effect-site TCI algorithm with plasma targeting within small range of target*

Description

Modified effect-site TCI algorithm that switches to plasma-targeting when the plasma concentration is within 20% of the target and the effect-site concentration is within 0.5% of the target. The modification decreases computation time and prevents oscillatory behavior in the effect-site concentrations.

Usage

```
tci_comb(Ct, pkmod, cptol = 0.2, cetol = 0.05, cp_cmpt = 1, ce_cmpt = 4, ...)
```

Arguments

Ct	Numeric vector of target effect-site concentrations.
pkmod	PK model
cptol	Percentage of plasma concentration required to be within to switch to plasma targeting.
cetol	Percentage of effect-site concentration required to be within to switch to plasma targeting.
cp_cmpt	Position of central compartment. Defaults to first compartment.
ce_cmpt	Position of effect-site compartment. Defaults to fourth compartment.
...	Arguments passed on to 'tci_plasma' and 'tci_effect' functions.

Value

Returns a numeric value with length one.

tci_effect — *TCI algorithm for effect-site targeting*

Description

Function for calculating a TCI infusion schedule corresponding to a set of target concentrations. This function makes use of formulas described by Shafer and Gregg (1992) in "Algorithms to rapidly achieve and maintain stable drug concentrations at the site of drug effect with a computer-controlled infusion pump"

Usage

```
tci_effect(
  Cet,
  pkmod,
  dtm = 1/6,
  ecmpt = NULL,
  tmax_search = 10,
  maxrt = 1200,
  grid_len = 1200,
  ...
)
```

Arguments

Cet	Numeric vector of target effect-site concentrations.
pkmod	PK model
dtm	Frequency of TCI updates. Default is 1/6 minutes = 10 seconds.
ecmpt	Effect site compartment number
tmax_search	Outer bound on times searched to find a maximum concentration following an infusion of duration dtm. Defaults to 20 minutes. May need to be increased if a drug has a slow elimination rate.
maxrt	Maximum infusion rate of TCI pump. Defaults to 1200.
grid_len	Number of time points used to identify time of maximum concentration. Can be increased for more precision.
...	Arguments used by pkmod.

Value

Returns a numeric value with length one.

---

tci_pd *Function to extend TCI grid to a set of PD targets*

Description

Function to extend TCI grid to a set of PD targets

Usage

```
tci_pd(pdresp, tms, pkmod, pdmod, pars_pk, pars_pd, pdinv, ecmpt = NULL, ...)
```

Arguments

pdresp	PD targets to be passed on to the TCI algorithm.
tms	Times corresponding to each PD target
pkmod	PK model function
pdmod	PD model function
pars_pk	PK model parameters
pars_pd	PD model parameters
pdinv	PD inverse function
ecmpt	Number corresponding to effect-site compartment. Defaults to the last compartment.
...	Arguments to be passed on to 'tci'. These can include alternate TCI algorithms if desired.

Value

Returns a vector of numeric values.

---

tci_plasma *TCI algorithm for plasma targeting*

Description

TCI algorithm based on the algorithm described by Jacobs (1990).

Usage

```
tci_plasma(Cpt, pkmod, dtm, maxrt = 1200, cmpt = 1, ...)
```

Arguments

Cpt	Target plasma concentration
pkmod	PK model
dtm	Duration of the infusion
maxrt	Maximum infusion rate. Defaults to 200 ml/min in reference to the maximum infusion rate of 1200 ml/h permitted by existing TCI pumps (e.g. Anestfusor TCI program).
cmpt	Compartment into which infusions are administered. Defaults to the first compartment.
...	Arguments passed on to pkmod.

Value

Returns a numeric value with length one.

Index