Package 'nlmixr2plot'

September 18, 2024

Title Nonlinear Mixed Effects Models in Population PK/PD, Plot Functions

Version 3.0.0

Description Fit and compare nonlinear mixed-effects models in differential equations with flexible dosing information commonly seen in pharmacokinetics and pharmacodynamics (Almquist, Leander, and Jirstrand 2015 <doi:10.1007/s10928-015-9409-1>). Differential equation solving is by compiled C code provided in the 'rxode2' package (Wang, Hallow, and James 2015 <doi:10.1002/psp4.12052>). This package is for 'ggplot2' plotting methods for 'nlmixr2' objects.

License GPL (>= 3)

URL https://github.com/nlmixr2/nlmixr2plot, https://nlmixr2.github.io/nlmixr2plot/

BugReports https://github.com/nlmixr2/nlmixr2plot/issues/

Imports ggplot2 (>= 3.4.0), nlmixr2est, nlmixr2extra, rxode2, utils, vpc, xgxr

Suggests testthat (>= 3.0.0), dplyr, withr, nlmixr2data

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Matthew Fidler [aut, cre] (<https://orcid.org/0000-0001-8538-6691>), Bill Denney [ctb] (<https://orcid.org/0000-0002-5759-428X>), Wenping Wang [aut], Vipul Mann [aut]

Maintainer Matthew Fidler <matthew.fidler@gmail.com>

Repository CRAN

Date/Publication 2024-09-18 17:50:02 UTC

Contents

plot.nlmixr2AugPred 2
plot.nlmixr2FitData 3
traceplot 4
vpcPlot 5

Index **9**

`plot.nlmixr2AugPred` *Plot a nlmixr2 augPred object*

Description

Plot a nlmixr2 augPred object

Usage

```
## S3 method for class 'nlmixr2AugPred'
plot(x, y, ...)
```

Arguments

`x`	augPred object
`y`	ignored, used to mach plot generic
`...`	Other arguments (ignored)

Value

Nothing called for side effects

Examples

```
library(nlmixr2est)
## The basic model consiss of an ini block that has initial estimates
one.compartment <- function() {
  ini({
    tka <- 0.45 # Log Ka
    tcl <- 1 # Log Cl
    tv <- 3.45    # Log V
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  # and a model block with the error sppecification and model specification
  model({
    ka <- exp(tka + eta.ka)
```

```
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v * center
    cp = center / v
    cp ~ add(add.sd)
  })
}

## The fit is performed by the function nlmixr/nlmix2 specifying the model, data and estimate
fit <- nlmixr2est::nlmixr2(one.compartment, theo_sd,  est="saem", saemControl(print=0))

# augPred shows more points for the fit:

a <- nlmixr2est::augPred(fit)

# you can plot it with plot(augPred object)
plot(a)
```

`plot.nlmixr2FitData` *Plot a nlmixr2 data object*

Description

Plot some standard goodness of fit plots for the focei fitted object

Usage

```
## S3 method for class 'nlmixr2FitData'
plot(x, ...)
```

Arguments

`x`	a focei fit object
`...`	additional arguments (currently ignored)

Value

An `nlmixr2PlotList` object (a list of ggplot2 objects with easier plotting for all of them at the same time)

Author(s)

Wenping Wang & Matthew Fidler

Examples

```
library(nlmixr2est)
one.compartment <- function() {
  ini({
    tka <- 0.45
    tcl <- 1
    tv <- 3.45
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v * center
    cp = center / v
    cp ~ add(add.sd)
  })
}

## The fit is performed by the function nlmixr/nlmix2 specifying the model, data and estimate
fit <- nlmixr2(one.compartment, theo_sd,  est="saem", saemControl(print=0, nBurn = 10, nEm = 20))

# This shows many goodness of fit plots
plot(fit)
```

---

traceplot *Produce trace-plot for fit if applicable*

Description

Produce trace-plot for fit if applicable

Usage

```
traceplot(x, ...)

## S3 method for class 'nlmixr2FitCore'
traceplot(x, ...)
```

Arguments

x	fit object
...	other parameters

Value

Fit traceplot or nothing.

Author(s)

Rik Schoemaker, Wenping Wang & Matthew L. Fidler

Examples

```
library(nlmixr2est)
## The basic model consiss of an ini block that has initial estimates
one.compartment <- function() {
  ini({
    tka <- 0.45 # Log Ka
    tcl <- 1 # Log Cl
    tv <- 3.45    # Log V
    eta.ka ~ 0.6
    eta.cl ~ 0.3
    eta.v ~ 0.1
    add.sd <- 0.7
  })
  # and a model block with the error sppecification and model specification
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v * center
    cp = center / v
    cp ~ add(add.sd)
  })
}

## The fit is performed by the function nlmixr/nlmix2 specifying the model, data and estimate
fit <- nlmixr2(one.compartment, theo_sd,  est="saem", saemControl(print=0))

# This shows the traceplot of the fit (useful for saem)
traceplot(fit)
```

---

vpcPlot *VPC based on ui model*

Description

VPC based on ui model

Usage

```
vpcPlot(
  fit,
  data = NULL,
  n = 300,
  bins = "jenks",
  n_bins = "auto",
  bin_mid = "mean",
  show = NULL,
  stratify = NULL,
  pred_corr = FALSE,
  pred_corr_lower_bnd = 0,
  pi = c(0.05, 0.95),
  ci = c(0.05, 0.95),
  uloq = fit$dataUloq,
  lloq = fit$dataLloq,
  log_y = FALSE,
  log_y_min = 0.001,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  smooth = TRUE,
  vpc_theme = NULL,
  facet = "wrap",
  scales = "fixed",
  labeller = NULL,
  vpcdb = FALSE,
  verbose = FALSE,
  ...,
  seed = 1009,
  idv = "time",
  cens = FALSE
)

vpcPlotTad(..., idv = "tad")

vpcCensTad(..., cens = TRUE, idv = "tad")

vpcCens(..., cens = TRUE, idv = "time")
```

Arguments

fit	nlmixr2 fit object
data	this is the data to use to augment the VPC fit. By default is the fitted data, (can be retrieved by getData), but it can be changed by specifying this argument.
n	Number of VPC simulations
bins	either "density", "time", or "data", "none", or one of the approaches available in

	classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
n_bins	when using the "auto" binning method, what number of bins to aim for
bin_mid	either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
show	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
stratify	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
pred_corr	perform prediction-correction?
pred_corr_lower_bnd	lower bound for the prediction-correction
pi	simulated prediction interval to plot. Default is c(0.05, 0.95),
ci	confidence interval to plot. Default is (0.05, 0.95)
uloq	Number or NULL indicating upper limit of quantification. Default is NULL.
lloq	Number or NULL indicating lower limit of quantification. Default is NULL.
log_y	Boolean indicting whether y-axis should be shown as logarithmic. Default is FALSE.
log_y_min	minimal value when using log_y argument. Default is 1e-3.
xlab	label for x axis
ylab	label for y axis
title	title
smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet	either "wrap", "columns", or "rows"
scales	either "fixed" (default), "free_y", "free_x" or "free"
labeller	ggplot2 labeller function to be passed to underlying ggplot object
vpcdb	Boolean whether to return the underlying vpcdb rather than the plot
verbose	show debugging information (TRUE or FALSE)
...	Args sent to rxSolve
seed	an object specifying if and how the random number generator should be initialized
idv	Name of independent variable. For vpcPlot() and vpcCens() the default is "time" for vpcPlotTad() and vpcCensTad() this is "tad"
cens	is a boolean to show if this is a censoring plot or not. When cens=TRUE this is actually a censoring vpc plot (with vpcCens() and vpcCensTad()). When cens=FALSE this is traditional VPC plot (vpcPlot() and vpcPlotTad()).

Value

Simulated dataset (invisibly)

Author(s)

Matthew L. Fidler

Examples

```
one.cmt <- function() {
 ini({
   tka <- 0.45; label("Ka")
   tcl <- log(c(0, 2.7, 100)); label("Cl")
   tv <- 3.45; label("V")
   eta.ka ~ 0.6
   eta.cl ~ 0.3
   eta.v ~ 0.1
   add.sd <- 0.7; label("Additive residual error")
 })
 model({
   ka <- exp(tka + eta.ka)
   cl <- exp(tcl + eta.cl)
   v <- exp(tv + eta.v)
   linCmt() ~ add(add.sd)
 })
}

fit <-
  nlmixr2est::nlmixr(
    one.cmt,
    data = nlmixr2data::theo_sd,
    est = "saem",
    control = list(print = 0)
  )

vpcPlot(fit)
```

Index

getData, *6*

plot.nlmixr2AugPred, 2
plot.nlmixr2FitData, 3

rxSolve, *7*

traceplot, 4

vpcCens *(*vpcPlot*)*, 5
vpcCensTad *(*vpcPlot*)*, 5
vpcPlot, 5
vpcPlotTad *(*vpcPlot*)*, 5