Package 'BeeGUTS'

August 9, 2022

Title General Unified Threshold Model of Survival for Bees using Bayesian Inference

Version 1.0.0

Description Tools to calibrate, validate, and make predictions with the General Unified Threshold model of Survival adapted for Bee species. The model is presented in the publication from Baas, J., Goussen, B., Miles, M., Preuss, T.G., Roessing, I. (submitted) and is based on the GUTS framework Jager, T., Albert, C., Preuss, T.G. and Ashauer, R. (2011) <doi:10.1021/es103092a>. The authors are grateful to Bayer A.G. for its financial support.

License GPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.1

Biarch true

Depends R (>= 3.4.0)

Imports methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.1.1), data.table, tidyr, ggplot2, cowplot, dplyr, magrittr, utils, gridExtra, odeGUTS

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

URL https://github.com/bgoussen/BeeGUTS

BugReports https://github.com/bgoussen/BeeGUTS/issues

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation yes

Author Benoit Goussen [aut, cre] (<https://orcid.org/0000-0001-7204-7981>),
Liubov Zakharova [ctb],
Romoli Carlo [ctb],
Bayer AG [cph]

Maintainer Benoit Goussen <benoit.goussen@ibacon.com>

Repository CRAN

Date/Publication 2022-08-09 12:00:10 UTC

R topics documented:

BeeGUTS-package 2
betacyfluthrinChronic 3
concAC 4
concAO 4
concCst 5
criteriaCheck 6
dataGUTS 7
fitBeeGUTS 9
fitBetacyfluthrin_Chronic 11
LCx 12
LCx.beeSurvFit 12
plot.beeSurvData 13
plot.beeSurvFit 14
plot.beeSurvPred 15
plot.beeSurvValidation 16
plot.ppc 17
ppc 18
ppc.beeSurvFit 18
predict.beeSurvFit 19
summary.beeSurvFit 20
traceplot 21
validate 21
validate.beeSurvFit 22

Index **24**

`BeeGUTS-package` *'BeeGUTS' package; a package to perform GUTS modelling for Bee experiments.*

Description

Provide tools to analyse the survival toxicity tests performed for bee species. It can be used to fit a Toxicokinetic-Toxicodynamic (TKTD) model adapted for bee standard studies (acute oral, acute contact, and chronic oral studies). The TKTD model used is the General Unified Threshold model of Survival (GUTS).

The package follows the concept and assumptions presented in Baas et al (submitted)

References

Baas, J., Goussen, B., Miles, M., Preuss, T.G., Roessing, I. (submitted). BeeGUTS – new integrative TKTD model for honey bees approach moving from single point estimates of toxicity and exposure to a holistic link between exposure and effect.

Jager, T., Albert, C., Preuss, T.G. and Ashauer, R. (2011). General Unified Threshold model of Survival - a toxicokinetic-toxicodynamic framework for ecotoxicology. doi: 10.1021/es103092a

Jager, T. and Ashauer, R. (2018). Modelling survival under chemical stress. A comprehensive guide to the GUTS framework. Version 1.0 https://leanpub.com/guts_book

EFSA PPR Scientific Opinion (2018). Scientific Opinion on the state of the art of Toxicokinetic/Toxicodynamic (TKTD) effect models for regulatory risk assessment of pesticides for aquatic organisms. https://www.efsa.europa.eu/en/efsajournal/pub/5377

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

---

`betacyfluthrinChronic` *Survival datasets for* Honey bees *exposed to constant concentration of Betacyfluthrin for 10 days.*

Description

Survival datasets for *Honey bees* exposed to constant concentration of Betacyfluthrin for 10 days.

Usage

```
data(betacyfluthrinChronic)
```

Format

A list of class `beeSurvData` constructed by `dataGUTS` containing:

`survData` A data frame containing the survival information over time for five treatments and a control in a wide format.

`survData_long` A data frame containing the survival information over time for five treatments and a control in a long format.

`concData` A data frame containing the concentration information over time for five treatments and a control in a wide format.

`concData_long` A data frame containing the concentration information over time for five treatments and a control in a long format.

`unitData` A character string containing the units of the concentration data.

`typeData` A character string containing the type of data (here Chronic_Oral).

`beeSpecies` A character string containing the species of bee of interest (here Honey_Bee).

`concModel` A data frame containing the concentration information recalculated for the species of bee and test type of interest in a wide format.

`concModel_long` A data frame containing the concentration information recalculated for the species of bee and test type of interest in a long format.

References

Bayer data.

---

concAC *Recalculate the concentrations for the acute contact tests for bees*

Description

Recalculate the concentrations for the acute contact tests for bees

Usage

```
concAC(cExt, expTime, k_ca = 0.4, ...)
```

Arguments

cExt	The concentration applied
expTime	The duration of the experiment in days
k_ca	Contact availability rate (d-1), default is 0.4
...	Not used

Value

A data frame containing a column with the time points and a column with the recalculated concentrations

Examples

```
conc <- concAC(cbind(3.1, 4, 6, 8), 4)
```

---

concAO *Recalculate concentration for the acute oral tests for bees*

Description

Recalculate concentration for the acute oral tests for bees

Usage

```
concAO(cExt, cTime = 0.25, expTime, k_sr = 0.625, ...)
```

Arguments

cExt	A dataframe of concentrations at time 0 concentration applied
cTime	The duration of exposure in days, default is 0.25 d
expTime	The duration of the experiment in days
k_sr	Stomach release rate (d-1), default is 0.625
...	Not used

Value

A data frame containing a column with the time points and a column with the recalculated concentrations

Examples

```
conc <- concAO(cExt = cbind(3.5, 6, 8, 10), cTime = 0.25, expTime = 4)
```

---

concCst *Recalculate the concentrations for the chronic oral tests for bees from mg a.s./kg feed to μg/bee*

Description

Recalculate the concentrations for the chronic oral tests for bees from mg a.s./kg feed to μg/bee

Usage

```
concCst(cExt, f_rate = c(25), targConc = 1, cstConcCal = TRUE, ...)
```

Arguments

cExt	The concentration dataframe in mg a.s./kg feed
f_rate	A vector containing the feeding rates of the bees in mg/bee/day. If the vector is of size 1, the same feeding rate is used for all test conditions. If the vector is of size >1, it should be of the same size as the number of condition and one feeding rate must be provided per condition. Default is 25 mg/bee/day
targConc	A numerical scalar representing the unit of the target concentration amongst (default = 1) • 1 for μg a.s./bee • 2 for ng a.s./bee • 3 for mg a.s./bee
cstConcCal	Logical. Indicate if concentrations should be recalculated from mg a.s./kg feed to Xg/bee
...	Not used

Value

A data frame containing a column with the time points and a column with the recalculated concentrations

Examples

```
cExt <- data.frame(SurvivalTime = c(0,10), Control = c(0,0),
        T1 = c(1, 1), T2 = c(5, 5), Dataset = c(1, 1))
conc <- concCst(cExt, targConc = 2)
```

`criteriaCheck` *Computes PPC and NRMSE as defined in EFSA 2018*

Description

Computes PPC and NRMSE as defined in EFSA 2018

Usage

```
criteriaCheck(x)
```

Arguments

`x`	an object of class `beeSurvFit` or `beeSurvPred`

Value

The function returns a list with three items:

`PPC`	The criterion, in percent, compares the predicted median number of survivors associated to their uncertainty limits with the observed numbers of survivors. Based on experience, PPC resulting in more than 50% of the observations within the uncertainty limits indicate good model performance (EFSA 2018). A fit of 100% may hide too large uncertainties of prediction (so covering all data).
`NRMSE`	The criterion, in percent, is based on the classical root-mean-square error (RMSE), used to aggregate the magnitudes of the errors in predictions for various time-points into a single measure of predictive power. In order to provide a criterion expressed as a percentage, NRMSE is the normalised RMSE by the mean of the observations. EFSA (2018) recognised that a NRMSE of less than 50% indicates good model performance
`SPPE`	A list with the Survival Probability Prediction Error per dataset and condition. Each dataset is in a sublist.

@references EFSA PPR Scientific Opinion (2018) *Scientific Opinion on the state of the art of Toxicokinetic/Toxicodynamic (TKTD) effect models for regulatory risk assessment of pesticides for aquatic organisms* https://www.efsa.europa.eu/en/efsajournal/pub/5377

@example data(fitBetacyfluthrin_Chronic) out <- criteriaCheck(fitBetacyfluthrin_Chronic)

dataGUTS *Read and format the data for the BeeGUTS model*

Description

Read data from a text or csv file and recalculate the exposure profile depending on the type of experiment (acute oral, acute contact, chronic oral).

Usage

```
dataGUTS(
  file_location = NULL,
  test_type = NULL,
  bee_species = "Honey_Bee",
  NA_string = getOption("datatable.na.strings", "NA"),
  ...
)
```

Arguments

file_location	List of Locations of text files containing each two datasets, one for the survival data, and one for the concentration data. Both datasets must be included in the same file and contain the same number of column in the same order. The following columns must be included in the survival dataset: • Survival time \[d\]: a vector of time in days • Control A vector of number of survivors for the control • T1 - Tn A vector of number of survivors for the treatments T1 to Tn, one column per treatment. A line containing the Concentration unit must be included directly after the end of the last row of the survival data. The following columns must be included in the concentration dataset • Concentration time \[d\]: a vector of time in days. • Control A vector of concentrations for the control • T1 - Tn A vector of concentration for the treatments T1 to Tn, one column per treatment. For the Acute_Oral and Acute_Contact, only the initial exposure concentration at time 0 is required. See detail section for example
test_type	list of test types amongst "Acute_Oral", "Acute_Contact", and "Chronic_Oral" this list must have the same length of the list of file locations
bee_species	the bee type. At the moment only "Honey_Bee" is supported
NA_string	a character vector of strings which are to be interpreted as NA values
...	Optional arguments to be passed to the concentration reconstruction (e.g.

- k_sr = for the stomach release rate (d-1), default is 0.625,
- k_ca = contact availability rate (d-1), default is 0.4), or
- cTime = the duration of exposure in days for the acute oral tests, default is 0.25 d
- cstConcCal = logical, recalculate concentration in the Chronic_Oral test from mg a.s./kg feed to Xg/bee (default is TRUE)
- f_rate = numerical vector, feeding rate used in the concentration recalculation in the Chronic_Oral (default is 25 mg/bee/day)
- targConc = numerical scalar, target concentration unit in the recalculation in the Chronic_Oral, 1 for µg/bee, 2 for ng/bee, 3 for mg/bee (default is 1).

Details

The filename must begin with name of the chemical substance being tested and each word of the filename should be separated via an underscore '_'.

#' Example of formatting of the input file for a chronic oral study

Survival time [d]	Control	T1	T2	T3	T4	T5
0	120	120	120	120	120	120
1	120	118	117	112	115	94
2	120	118	115	112	98	88
3	120	118	114	106	83	27
4	119	118	113	103	67	9
5	119	118	112	100	43	3
Concentration unit: ug/bee/day						
Concentration time [d]	Control	T1	T2	T3	T4	T5
0	0	3	7	12	41	68
5	0	3	7	12	41	68

Value

An object of class beeSurvData, which is a list with the following information:

nDatasets	Number of files passed to the function
survData	A table containing the survival data as entered by the user in the input file
survData_long	A data frame containing the survival data in long format for model purposes
concData	A table containing the concentration data as entered by the user in the input file
concData_long	A data frame containing concentration data in long format
unitData	A character vector containing the units of the data as entered in the line Concentration unit of the input file
typeData	A character vector containing the type of experiment
beeSpecies	A character vector containing the type bee
concModel	A data frame containing the concentration data as recalculated by the model
concModel_long	A data frame containing the concentration data as recalculated by the model in a long format

Each element of the list is itself a list to account for multiple files that can be passed as input.

Examples

```
file_location <- system.file("extdata", "betacyfluthrin_chronic_ug.txt", package = "BeeGUTS")
lsData <- dataGUTS(file_location = c(file_location),
                   test_type = c('Chronic_Oral'), cstConcCal = FALSE)
```

`fitBeeGUTS` *Fit a GUTS model for bees survival analysis using Bayesian Inference (stan)*

Description

The function `fitBeeGUTS` estimates the parameters of a GUTS model for the stochastic death (SD) or individual tolerance (IT) death mechanisms for survival analysis using Bayesian inference.

Usage

```
fitBeeGUTS(
  data,
  modelType = NULL,
  distribution = "loglogistic",
  priorsList = NULL,
  parallel = TRUE,
  nCores = parallel::detectCores() - 1L,
  nChains = 3,
  nIter = 2000,
  nWarmup = floor(nIter/2),
  thin = 1,
  adaptDelta = 0.95,
  odeIntegrator = "rk45",
  relTol = 1e-08,
  absTol = 1e-08,
  maxSteps = 1000,
  ...
)
```

Arguments

`data`	An object of class `beeSurvData`
`modelType`	A model type between `"SD"` for Stochastic Death and `"IT"` for Individual Tolerance.
`distribution`	A distribution for the IT death mechanism. To be chosen between `"loglogistic"` and `"lognormal"`. Default is `"loglogistic"`
`priorsList`	A list containing the prior distribution for the parameter considered. By default, when no priors are provided (default is `NULL`), priors are set automatically based on the experimental design (adapted from Delignette-Muller et al 2017)

parallel	Logical indicating whether parallel computing should be used or not. Default is TRUE
nCores	A positive integer specifying the number of cores to use. Default is one core less than maximum number of cores available
nChains	A positive integer specifying the number of MCMC chains to run. Default is 3.
nIter	A positive integer specifying the number of iteration to monitor for each MCMC chain. Default is 2000
nWarmup	A positive integer specifying the number of warmup iteration per chain. Default is half the number of iteration
thin	A positive integer specifying the interval between the iterations to monitor. Default is 1 (all iterations are monitored)
adaptDelta	A double, bounded between 0 and 1 and controlling part of the sampling algorithms. See the control in the function stan rstan::stan() of the package rstan. The default is 0.95.
odeIntegrator	A string specifying the integrator used to solve the system of differential equations (ODE) in the stan module. To be chosen between "rk45" and "bdf". Default is "rk45".
relTol	A double, bounded between 0 and 1 and controlling the relative tolerance of the accuracy of the solutions generated by the integrator. A smaller tolerance produces more accurate solution at the expanse of the computing time. Default is 1e-8
absTol	A double, bounded between 0 and 1 and controlling the absolute tolerance of the accuracy of the solutions generated by the integrator. A smaller tolerance produces more accurate solution at the expanse of the computing time. Default is 1e-8
maxSteps	A double controlling the maximum number of steps that can be taken before stopping a runaway simulation. Default is 1000
...	Additional parameters to be passed to sampling from stan

Details

The automated prior determination is modified from Delignette-Muller et al. by considering that the minimal concentration for the prior can be close to 0 (1e-6) whereas the original paper considered the lowest non-zero concentration. Similarly, the minimal kd considered for the prior calculation was reduced to allow more chance to capture slow kinetics.

Value

The function fitBeeGUTS returns the parameter estimates of the General Unified Threshold model of Survival (GUTS) in an object of class beeSurvFit. This object is a list composed of the following:

stanFit	An object of S4 class stanfit. More information is available in the package rstan.
data	The data object provided as argument of the function

dataFit	A list of data passed to the Stan model object
setupMCMC	A list containing the setup used for the MCMC chains
modelType	A character vector specifying the type of GUTS model used between SD and IT
distribution	A character vector specifying the type of distribution used in case IT was used; NA otherwise
messages	A character vector containing warning messages

References

Delignette-Muller, M.L., Ruiz P. and Veber P. (2017). Robust fit of toxicokinetic-toxicodynamic models using prior knowledge contained in the design of survival toxicity tests. doi: 10.1021/acs.est.6b05326

Examples

```
data(betacyfluthrinChronic)
fit <- fitBeeGUTS(betacyfluthrinChronic, modelType = "SD", nIter = 1000, nCores = 2)
```

---

fitBetacyfluthrin_Chronic

Model calibration results datasets for Honey bees *exposed to constant concentration of Betacyfluthrin for 10 days.*

Description

Model calibration results datasets for *Honey bees* exposed to constant concentration of Betacyfluthrin for 10 days.

Usage

```
data(fitBetacyfluthrin_Chronic)
```

Format

A list of class beeSurvFit constructed by fitBeeGUTS containing:

stanFit A 'stanfit' object containing the results of the calibration.

data A 'beeSurvData' objects with the user data used for the calibration.

dataFit A list containing the priors and data formatted for the calibration algorithm.

setupMCMC A list containing the setup used for the MCMC.

modelType A character string containing the type of GUTS model used (here 'SD').

distribution A character string containing the distribution used (IT only, here 'NA').

messages A character string containing the error messages if Rhat >1.1 (here 'NA').

References

Bayer data.

`LCx` *Predict the Lethal Concentration method at which x\ specified time-point for a* `beeSurvFit` *object*

Description

Predict median and 95\

Usage

```
LCx(object, ...)
```

Arguments

`object`	An object used to select a method
`...`	Further arguments to be passed to generic methods

Details

When class of `object` is `beeSurvFit`, see LCx.beeSurvFit.

Value

A `LCx` object containing the results of the lethal concentration predictions

---

`LCx.beeSurvFit` *Predict the Lethal Concentration at which x\ specified time-point for a* `beeSurvFit` *object*

Description

Predict the Lethal Concentration at which x\ specified time-point for a `beeSurvFit` object

Usage

```
## S3 method for class 'beeSurvFit'
LCx(
  object,
  X = 50,
  testType = "Chronic_Oral",
  timeLCx = NULL,
  concRange = NULL,
  nPoints = 100,
  ...
)
```

Arguments

`object`	An object of class `beeSurvFit`
`X`	Percentage of individuals dying (e.g., 50 for the LC_{50})
`testType`	Test type for which the LC_X is calculated amongst "Acute_Oral", "Acute_Contact", and "Chronic_Oral". Note that for "Acute_Oral" and "Acute_Contact", the concentration will be reconstructed as in the dataGUTS function (not recommended as this might not make sense for LC_X estimations. Default is "Chronic_Oral"
`timeLCx`	A scalar giving the time at which LC_x is predicted. If `NULL`, the latest time point of the experiment used in the calibration is used
`concRange`	A vector of length 2 with minimal and maximal value of the range of concentration. If `NULL`, the range is define between 0 and the highest tested concentration of the calibration experiment.
`nPoints`	Number of time point in `concRange` between 0 and the maximal concentration. 100 by default.
`...`	Further arguments to be passed to generic methods

Value

A `LCx` object containing the results of the lethal concentration predictions

Examples

```
data(fitBetacyfluthrin_Chronic)
out <- LCx(fitBetacyfluthrin_Chronic)
```

`plot.beeSurvData` *Plotting method for* `beeSurvData` *objects*

Description

This is the generic `plot` S3 method for the `beeSurvData` class. It plots the number of survivors as a function of time as well as the reconstructed concentrations for `"Acute_Oral"` and `"Acute_Contact"` test types.

Usage

```
## S3 method for class 'beeSurvData'
plot(
  x,
  ...,
  xlab = "Time [d]",
  ylab1 = "Number of survivors",
  ylab2 = "Concentration",
  main = paste("Data from a", x$typeData, "test on", x$beeSpecies)
)
```

Arguments

x	An object of class beeSurvData
...	Additional parameters to generic plot function (not used)
xlab	A character string for the label of the x-axis
ylab1	A character string for the label of the y-axis of the survivor plots
ylab2	A character string for the label of the y-axis of the concentration plots
main	A character string for the title label plot

Value

A graphic with the input data

Examples

```
data(betacyfluthrinChronic)
plot(betacyfluthrinChronic)
```

---

plot.beeSurvFit *Plotting method for* beeSurvFit *objects*

Description

This is the generic plot S3 method for the beeSurvFit class. It plots the number of survivors as a function of time as well as the reconstructed concentrations for "Acute_Oral" and "Acute_Contact" test types.

Usage

```
## S3 method for class 'beeSurvFit'
plot(
  x,
  ...,
  xlab = "Time [d]",
  ylab1 = "Number of survivors",
  ylab2 = "Concentration",
  main = paste("Calibration results for a", x$data$typeData, "test on",
    x$data$beeSpecies)
)
```

Arguments

x	An object of class beeSurvFit
...	Additional parameters to generic plot functions (not used)
xlab	A character string for the label of the x-axis
ylab1	A character string for the label of the y-axis of the survivor plots
ylab2	A character string for the label of the y-axis of the concentration plots
main	A character string for the title label plot

Value

A graphic with the results of the fit

Examples

```
data(fitBetacyfluthrin_Chronic)
plot(fitBetacyfluthrin_Chronic)
```

---

plot.beeSurvPred *Plotting method for* beeSurvPred *objects*

Description

This is the generic plot S3 method for the beeSurvPred class. It plots the predicted number of survivors for the exposure concentration entered by the user.

Usage

```
## S3 method for class 'beeSurvPred'
plot(
  x,
  ...,
  xlab = "Time [d]",
  ylab1 = "Survival probability",
  ylab2 = "Concentration",
 main = paste("Predictions results for a BeeGUTS", x$modelType, "calibrated for",
    x$beeSpecies)
)
```

Arguments

x	An object of class beeSurvPred
...	Additional parameters to generic plot functions (not used)
xlab	A character string for the label of the x-axis
ylab1	A character string for the label of the y-axis of the survivor plots
ylab2	A character string for the label of the y-axis of the concentration plots
main	A character string for the title label plot

Value

A graphic with results of the forward prediction

Examples

```
dataPredict <- data.frame(time = c(1:10, 1:10, 1:10),
                    conc = c(rep(5, 10), rep(10, 10), rep(15, 10)),
                    replicate = c(rep("rep1", 10), rep("rep2", 10), rep("rep3", 10)),
                    NSurv = c(rep(5, 10), rep(10, 10), rep(15, 10)))
data(fitBetacyfluthrin_Chronic)
prediction <- predict(fitBetacyfluthrin_Chronic, dataPredict)
plot(prediction)
```

---

plot.beeSurvValidation

Plotting method for beeSurvValidation *objects*

Description

This is the generic plot S3 method for the beeSurvValid class. It plots the number of survivors as a function of time as well as the reconstructed concentrations for "Acute_Oral" and "Acute_Contact" test types.

Usage

```
## S3 method for class 'beeSurvValidation'
plot(
  x,
  ...,
  xlab = "Time [d]",
  ylab1 = "Number of survivors",
  ylab2 = "Concentration",
  main = paste("Validation results for a BeeGUTS", x$typeData, "calibrated for",
    x$beeSpecies)
)
```

Arguments

x	An object of class beeSurvValid
...	Additional parameters to generic plot functions (not used)
xlab	A character string for the label of the x-axis
ylab1	A character string for the label of the y-axis of the survivor plots
ylab2	A character string for the label of the y-axis of the concentration plots
main	A character string for the title label plot

Value

A graphic with the results of the validation

Examples

```
data(betacyfluthrinChronic) # Load dataset for validation
data(fitBetacyfluthrin_Chronic)
validation <- validate(fitBetacyfluthrin_Chronic, betacyfluthrinChronic)
plot(validation)
```

`plot.ppc` *Plotting method for* ppc *objects*

Description

Plotting method for ppc objects

Usage

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

Arguments

x	An object of class ppc.
...	Further arguments to be passed to generic methods.

Value

an object of class ggplot.

Examples

```
data(fitBetacyfluthrin_Chronic)
out <- ppc(fitBetacyfluthrin_Chronic)
plot(out)
```

ppc — *Generates an object to be used in posterior predictive check for* beeSurvFit, beeSurvPred

Description

Generates an object to be used in posterior predictive check for beeSurvFit, beeSurvPred

Usage

```
ppc(x)
```

Arguments

x	an object used to select a method ppc

Value

a data.frame of class ppc

ppc.beeSurvFit — *Posterior predictive check method for* beeSurvFit *objects*

Description

Posterior predictive check method for beeSurvFit objects

Usage

```
## S3 method for class 'beeSurvFit'
ppc(x)
```

Arguments

x	an object of class beeSurvFit

Value

a data.frame of class ppc

Examples

```
data(fitBetacyfluthrin_Chronic)
out <- ppc(fitBetacyfluthrin_Chronic)
```

predict.beeSurvFit *Predict method for* beeSurvFit *objects*

Description

This is the generic predict S3 method for the beeSurvFit class. It predict the survival over time for the concentration profiles entered by the user. No concentration reconstructions are performed here. Functions odeGUTS::predict_ode() from the morse package is used. This might be changed in a future update

Usage

```
## S3 method for class 'beeSurvFit'
predict(object, dataPredict, ...)
```

Arguments

object	An object of class beeSurvFit
dataPredict	Data to predict in the format as a dataframe containing the following column: • time: A vector of time in days • conc: A vector of number of survivors of same length • replicate A vector replicate name
...	Additional arguments to be parsed to the predict.survFit method from odeGUTS (e.g. mcmc_size = 1000 is to be used to reduce the number of mcmc samples in order to speed up the computation. mcmc_size is the number of selected iterations for one chain. Default is 1000. If all MCMC is wanted, set argument to NULL. hb_value = FALSE the background mortality hb is set to a fixed value. If TRUE, parameter hb taken from the posterior (only works if one hb value was estimated. The default is FALSE. hb_valueFORCED = 0 hb_valueFORCED If hb_value is FALSE, it fix hb. The default is 0

Value

A beeSurvPred object containing the results of the forwards prediction

Examples

```
dataPredict <- data.frame(time = c(1:5, 1:15),
                          conc = c(rep(5, 5), rep(15, 15)),
                          replicate = c(rep("rep1", 5), rep("rep2", 15)))
data(fitBetacyfluthrin_Chronic)
prediction <- predict(fitBetacyfluthrin_Chronic, dataPredict)
```

summary.beeSurvFit *Summary of* beeSurvFit *objects*

Description

This is the generic summary S3 method for the beeSurvFit class. It shows the quantiles of priors and posteriors on parameters.

This is the generic summary S3 method for the LCx class. It shows the median and 95% credible interval of the calculated LCx.

Usage

```
## S3 method for class 'beeSurvFit'
summary(object, ...)

## S3 method for class 'beeSurvFit'
summary(object, ...)
```

Arguments

object	An object of class LCx
...	Additional arguments to be parsed to the generic summary method (not used)

Value

A summary of the beeSurvFit object

A summary of the LCx object

Examples

```
data(fitBetacyfluthrin_Chronic)
summary(fitBetacyfluthrin_Chronic)


data(fitBetacyfluthrin_Chronic)
out <- LCx(fitBetacyfluthrin_Chronic)
summary(out)
```

traceplot *Plotting method for traces and densities for* beeSurvFit *objects*

Description

This is the generic traceplot S3 method for the beeSurvFit class. It plots the traces with as well as the densities for the parameters of the GUTS IT or GUTS SD. The traceplot includes by default the warmup iterations, the density plot does not include them

Usage

```
traceplot(object, ..., incWarmup_trace = TRUE, incWarmup_dens = FALSE)

## S3 method for class 'beeSurvFit'
traceplot(object, ..., incWarmup_trace = TRUE, incWarmup_dens = FALSE)
```

Arguments

object	An object of class beeSurvFit to be plotted
...	Additional parameters to be parsed to generic rstan plot functions
incWarmup_trace	A logical indicating whether the warmup iterations should be plotted in the traceplot (default TRUE)
incWarmup_dens	A logical indicating whether the warmup iterations should be plotted in the density plot (default FALSE)

Value

A graphic with the traceplots and densities of the fit

Examples

```
data(fitBetacyfluthrin_Chronic)
traceplot(fitBetacyfluthrin_Chronic)
```

validate *Validation method for* beeSurvFit *objects*

Description

This is a validation method for the beeSurvFit object. It perform forwards predictions for a specific concentration profile and compare these prediction to the respective experimental data.

Usage

```
validate(object, dataValidate, ...)
```

Arguments

object	An object of class beeSurvFit
dataValidate	Data to validate in the format of the experimental data used for fit (dataGUTS)
...	Additional arguments to be parsed to the predict.survFit method from odeGUTS (e.g. mcmc_size = 1000 is to be used to reduce the number of mcmc samples in order to speed up the computation. mcmc_size is the number of selected iterations for one chain. Default is 1000. If all MCMC is wanted, set argument to NULL., hb_value = FALSE the background mortality hb is taken into account from the posterior. If FALSE, parameter hb is set to a fixed value. The default is FALSE. hb_valueFORCED = 0 hb_valueFORCED If hb_value is FALSE, it fix hb. The default is 0

Value

An object of class beeSurvValidation.

---

validate.beeSurvFit *Validate method for* beeSurvFit *objects*

Description

This is the generic validate S3 method for the beeSurvFit class. It predict the survival over time for the concentration profiles entered by the user.

Usage

```
## S3 method for class 'beeSurvFit'
validate(object, dataValidate, ...)
```

Arguments

object	An object of class beeSurvFit
dataValidate	Data to validate in the format of the experimental data used for fit (dataGUTS)
...	Additional arguments to be parsed to the predict.survFit method from odeGUTS (e.g. mcmc_size = 1000 is to be used to reduce the number of mcmc samples in order to speed up the computation. mcmc_size is the number of selected iterations for one chain. Default is 1000. If all MCMC is wanted, set argument to NULL., hb_value = FALSE the background mortality hb is taken into account from the posterior. If FALSE, parameter hb is set to a fixed value. The default is FALSE. hb_valueFORCED = 0 hb_valueFORCED If hb_value is FALSE, it fix hb. The default is 0

Value

A beeSurvValidation object with the results of the validation

Examples

```
data(betacyfluthrinChronic)
data(fitBetacyfluthrin_Chronic)
validation <- validate(fitBetacyfluthrin_Chronic, betacyfluthrinChronic)
```

Index

∗ **dataset**
betacyfluthrinChronic, 3
fitBetacyfluthrin_Chronic, 11

BeeGUTS *(*BeeGUTS-package*)*, 2
BeeGUTS-package, 2
betacyfluthrinChronic, 3

concAC, 4
concAO, 4
concCst, 5
criteriaCheck, 6

dataGUTS, 7, *13*

fitBeeGUTS, 9
fitBetacyfluthrin_Chronic, 11

LCx, 12
LCx.beeSurvFit, *12*, 12

odeGUTS::predict_ode(), *19*

plot.beeSurvData, 13
plot.beeSurvFit, 14
plot.beeSurvPred, 15
plot.beeSurvValidation, 16
plot.ppc, 17
ppc, 18
ppc.beeSurvFit, 18
predict.beeSurvFit, 19

rstan::stan(), *10*

summary.beeSurvFit, 20

traceplot, 21

validate, 21
validate.beeSurvFit, 22