Package: BeeGUTS (via r-universe)

October 29, 2026

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

Version 1.5.0

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. (2022) <doi:10.1002/etc.5423> and Baas, J., Goussen, B., Taenzler, V., Roeben, V., Miles, M., Preuss, T.G., van den Berg, S., Roessink, I. (2024) <doi:10.1002/etc.5871>, 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 7.2.3

Biarch true

NeedsCompilation yes

Depends R (>= 3.5.0)

Imports methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), rstantools (>= 2.1.1), data.table, tidyr, ggplot2, cowplot, dplyr, magrittr, utils, gridExtra, odeGUTS, parallel, doParallel, foreach

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

SystemRequirements GNU make

URL https://github.com/ibacon-GmbH-Modelling/BeeGUTS

BugReports https://github.com/ibacon-GmbH-Modelling/BeeGUTS/issues

Suggests rmarkdown, knitr

VignetteBuilder knitr

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

Maintainer Carlo Romoli <carlo.romoli@ibacon.com>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-01-15 17:40:35 UTC

RemoteUrl https://github.com/cran/BeeGUTS

RemoteRef HEAD

RemoteSha 3aab55f523d9b96e98dec07d138b01273b0f9d49

Contents

BeeGUTS-package . . . 3
betacyfluthrinChronic . . . 4
concAC . . . 5
concAO . . . 5
concCst . . . 6
correlation_plot . . . 7
criteriaCheck . . . 7
dataGUTS . . . 8
fitBeeGUTS . . . 10
fitBetacyfluthrin_Chronic . . . 13
LCx . . . 14
LCx.beeSurvFit . . . 14
plot.beeSurvData . . . 15
plot.beeSurvFit . . . 16
plot.beeSurvPred . . . 17
plot.beeSurvValidation . . . 18
plot.ppc . . . 19
ppc . . . 19
ppc.beeSurvFit . . . 20
ppc.beeSurvValidation . . . 20
predict.beeSurvFit . . . 21
predict2 . . . 22
priorposterior_plot . . . 23
ShortTimeEffects . . . 24
ShortTimeEffects.beeSurvFit . . . 25
summary.beeSurvFit . . . 26
summary.LCx . . . 26
traceplot . . . 27
TRT . . . 28

TRT.beeSurvFit 28
valBetacyfluthrinChronic 29
validate 30
validate.beeSurvFit 31

Index **32**

`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. (2022)

References

Baas, J., Goussen, B., Taenzler, V., Roeben, V., Miles, M., Preuss, T.G., van den Berg, S. and Roessink, I. (2024), Comparing Sensitivity of Different Bee Species to Pesticides: A TKTD modeling approach. Environ Toxicol Chem, 43: 1431-1441. doi:10.1002/etc.5871

Baas, J., Goussen, B., Miles, M., Preuss, T.G., Roessing, I. (2022). BeeGUTS—A Toxicokinetic–Toxicodynamic Model for the Interpretation and Integration of Acute and Chronic Honey Bee Tests. doi:10.1002/etc.5423

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

EFSA (European Food Safety Authority), Adriaanse P, Arce A, Focks A, Ingels B, Jölli D, Lambin S, Rundlöf M, Süßenbach D, Del Aguila M, Ercolano V, Ferilli F, Ippolito A, Szentes Cs, Neri FM, Padovani L, Rortais A, Wassenberg J and Auteri D, (2023). Revised guidance on the risk assessment of plant protection products on bees (Apis mellifera, Bombus spp. and solitary bees). EFSA Journal doi:10.2903/j.efsa.2023.7989

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:

nDatasets An integer representing the number of datasets used.

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.

messages A data frame containing the warning messages returned by the function.

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, binning = 0.5)
```

Arguments

cExt	The concentration applied
expTime	The duration of the experiment in days
k_ca	Contact availability rate (d-1), default is 0.4
binning	binning of the time vector, default 0.5 d

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, binning = 0.5)
```

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
binning	binning of the time vector, default 0.5 d

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)
```

correlation_plot *Plotting of correlation plots for* beeSurvFit *objects*

Description

It plots the correlations between the parameters of the GUTS IT or GUTS SD. The plots do not include warmup iterations

Usage

```
correlation_plot(object)
```

Arguments

object	An object of class beeSurvFit to be plotted

Value

A graphic with the plot of the correlations between parameters or a a list of graphics in case multiple datasets are present. If a list is returned, the first graphic is the correlation between hb values and the second is the correlation between all other parameters

Examples

```
data(fitBetacyfluthrin_Chronic)
correlation_plot(fitBetacyfluthrin_Chronic)
```

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://efsa.onlinelibrary.wiley.com/doi/10.2903/j.efsa.2018.5377

@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 among "Honey_Bee", "Bumble_Bee", "Osmia_bicornis", and "User_Bee". If "User_Bee" is selected, optional arguments to be passed to the concentration reconstruction need to be defined.

`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 for Honey bee,
- `k_ca` = contact availability rate (d-1), default is 0.4 for Honey bee), 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 for honey bee)
- `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).
- `binning` = binning of the time vector of the exposure for acute tests (Acute_Oral and Acute_Contact), default 0.5 d

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'),
                   bee_species = "Honey_Bee",
                   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 "euler". Default is "rk45". The "euler" integrator uses a fixed step of 0.01 d to calculate the solution (avoid using it for fast varying concentrations)
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 $x\%$ of organisms die for any specified time-point for a* `beeSurvFit` *object*

Description

Predict median and 95% credible interval of the $x\%$ Lethal Concentration.

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 $x\%$ of organisms die for any specified time-point for a* `beeSurvFit` *object*

Description

Predict the Lethal Concentration at which $x\%$ of organisms die for any 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,
  ncores = NULL,
  ...
)
```

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.
ncores	Number of cores for parallelization
...	Further arguments to be passed to generic methods

Value

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

Examples

```
data(fitBetacyfluthrin_Chronic)
out <- LCx(fitBetacyfluthrin_Chronic,ncores=2)
```

---

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 of a BeeGUTS model calibrated for", x$beeSpecies, "on a ",
    x$typeData, "for", x$beeSpeciesVal)
)
```

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)
```

---

ppc.beeSurvValidation *Posterior predictive check method for* beeSurvValidation *objects*

Description

Posterior predictive check method for beeSurvValidation objects

Usage

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

Arguments

x	an object of class beeSurvValidation

Value

a data.frame of class ppc

Examples

```
data(fitBetacyfluthrin_Chronic)
data(betacyfluthrinChronic)
valid <- validate(fitBetacyfluthrin_Chronic,betacyfluthrinChronic)
out <- ppc(valid)

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

---

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 (If more than one dataset is present, use the optional argument hb_dataset to specify which hb to use). 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)
```

---

predict2 *Predict method for* beeSurvFit *objects*

Description

This is an updated predict 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

```
predict2(
  object,
  dataPredict,
  userhb_value = 0,
  calib_hb = FALSE,
  ndatahb = 1,
  ...
)
```

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

userhb_value	User defined background mortality rate parameter. If a single value is provided, a single fixed value is used. If an array of two elements is give, the first element is the hb value and the second element is the approximate value of the desired 97.5% upperlimit on hb.
calib_hb	Logical argument. If TRUE uses for the predictions the background mortality value of the calibration dataset. Default value is FALSE.
ndatahb	Used in combination with calib_hb=TRUE in case the calibration object has multiple datasets. The value dictates which hb value to use (between the different dataset).
...	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.

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 <- predict2(fitBetacyfluthrin_Chronic, dataPredict)
```

---

priorposterior_plot *Plotting of prior and posterior plots for* beeSurvFit *objects*

Description

It plots the comparison between the prior and the posterior distributions of the parameters of the GUTS IT or GUTS SD models. The plots do not include warmup iterations

Usage

```
priorposterior_plot(object)
```

Arguments

object	An object of class beeSurvFit to be plotted

Value

A list of graphic objects with the plot of the comparison between prior and posterior distribution of the parameters. The first entry of the list is dedicated to the hb value (multiple plots in case of multiple datasets), the second element of the list is dedicated to the other model parameters

Examples

```
data(fitBetacyfluthrin_Chronic)
priorposterior_plot(fitBetacyfluthrin_Chronic)
```

`ShortTimeEffects` *Perform comparison between LLD50 at 2 days and at 10 days as for EFSA revised guideline (2023) - Section 6.6*

Description

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

Usage

```
ShortTimeEffects(
  object,
  fullcalculation = FALSE,
  concRange = NULL,
  nPoints = NULL,
  ncores = NULL
)
```

Arguments

`object`	An object used to select a method
`fullcalculation`	Compute the LDD50 from day 1 to day 10 of the Chronic test. This can increase the computation time
`concRange`	Argument of LCx, range of concentrations to find LDD50
`nPoints`	Argument of LCx, Number of time point in `concRange` between 0 and the maximal concentration. 100 by default.
`ncores`	Argument of LCx, number of cores for parallelization

Details

Copyright 2024 C. Romoli, ibacon GmbH

Value

A object of class `ggplot` containing the graph of the comparison between LDD50 at day 2 and day 10 and the data.frame with the plotted values.

ShortTimeEffects.beeSurvFit *Calculate the presence of Time Reinforced Toxicity for the compound from the calibrated model* beeSurvFit *object.*

Description

Calculate the presence of Time Reinforced Toxicity for the compound from the calibrated model beeSurvFit object.

Usage

```
## S3 method for class 'beeSurvFit'
ShortTimeEffects(
  object,
  fullcalculation = FALSE,
  concRange = NULL,
  nPoints = NULL,
  ncores = NULL
)
```

Arguments

object	An object of class beeSurvFit
fullcalculation	Compute the LDD50 from day 1 to day 10 of the Chronic test. This can increase the computation time
concRange	Argument of LCx, range of concentrations to find LDD50
nPoints	Argument of LCx, Number of time point in concRange between 0 and the maximal concentration. 100 by default.
ncores	Argument of LCx, number of cores for parallelization

Value

A object of class ggplot containing the graph of the comparison between LDD50 at day 2 and day 10 and the data.frame with the plotted values.

Examples

```
data(fitBetacyfluthrin_Chronic)
ShortTimeEffects(fitBetacyfluthrin_Chronic, ncores=2)
```

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.

Usage

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

Arguments

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

Value

A summary of the beeSurvFit object

Examples

```
data(fitBetacyfluthrin_Chronic)
summary(fitBetacyfluthrin_Chronic)
```

summary.LCx *Summary of* LCx *objects*

Description

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 'LCx'
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 LCx object

Examples

```
data(fitBetacyfluthrin_Chronic)
out <- LCx(fitBetacyfluthrin_Chronic,ncores=2)
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)
```

TRT *Perform the extrapolation of the LDD50 to 27 days and check the presence of reinforced toxicity as for EFSA revised guideline (2023) - Annex G*

Description

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

Usage

```
TRT(object, concRange = NULL, ncores = NULL)
```

Arguments

`object`	An object used to select a method
`concRange`	Argument of LCx, range of concentrations to find LDD50
`ncores`	Argument of LCx, number of cores for parallelization

Details

Copyright 2023-2024 C. Romoli, ibacon GmbH

Value

A `ggplot` object with graph of the LDD extrapolation compared to the Haber's law and a data.frame with the calculations

TRT.beeSurvFit *Calculate the presence of Time Reinforced Toxicity for the compound from the calibrated model* `beeSurvFit` *object.*

Description

Calculate the presence of Time Reinforced Toxicity for the compound from the calibrated model `beeSurvFit` object.

Usage

```
## S3 method for class 'beeSurvFit'
TRT(object, concRange = NULL, ncores = NULL)
```

Arguments

object	An object of class beeSurvFit
concRange	Argument of LCx, range of concentrations to find LDD50
ncores	Argument of LCx, number of cores for parallelization

Value

A object of class ggplot containing the graph of the comparison between Haber's law and the predicted lethal doses at 10 and 27 days and a data.frame with the plotted values.

Examples

```
data(fitBetacyfluthrin_Chronic)
TRT(fitBetacyfluthrin_Chronic,ncores=2)
```

---

valBetacyfluthrinChronic

Pre-calculated validation results for Honey bees *exposed to constant concentration of Betacyfluthrin for 10 days. This is a toy example that uses the calibrated model of the object fitBetacyfluthrin_Chronic and the dataset betacyfluthrinChronic*

Description

Pre-calculated validation results for *Honey bees* exposed to constant concentration of Betacyfluthrin for 10 days. This is a toy example that uses the calibrated model of the object fitBetacyfluthrin_Chronic and the dataset betacyfluthrinChronic

Usage

```
data(valBetacyfluthrinChronic)
```

Format

A list of class beeSurvFit constructed by fitBeeGUTS containing:

parsPost A data.frame with the summary of the posterior distribution of the model parameters.

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

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

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

beeSpeciesVal A character string containing the species of bee of the validation dataset (here Honey_Bee).

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

setupMCMC A list containing the setup used for the MCMC.

`sim` A data.frame with the results of the model prediction.

`EFSA` A list containing the results of goodness of fit checks as defined by EFSA.

`data` A data.frame with the user data used for the validation.

`dataModel` A data.frame with the data that were used for the model calibration.

`hbfit` NULL

References

Bayer data.

`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, fithb = FALSE, ...)
```

Arguments

`object`	An object of class `beeSurvFit`
`dataValidate`	Data to validate in the format of the experimental data used for fit (dataGUTS)
`fithb`	Logical argument. If `TRUE`, control data of the validation experiment are fitted to estimate the value of the background mortality rate. If `FALSE`, background mortality can be fixed with the optional argument `hb_valueFORCED`
`...`	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_valueFORCED = 0` hb_valueFORCED If `fithb` 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, fithb = FALSE, ...)
```

Arguments

object	An object of class beeSurvFit
dataValidate	Data to validate in the format of the experimental data used for fit (dataGUTS)
fithb	Logical argument. If TRUE, control data of the validation experiment are fitted to estimate the value of the background mortality rate. If FALSE, background mortality can be fixed with the optional argument hb_valueFORCED
...	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_valueFORCED = 0 hb_valueFORCED If fithb 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, 4
fitBetacyfluthrin_Chronic, 13
valBetacyfluthrinChronic, 29

BeeGUTS *(*BeeGUTS-package*)*, 3
BeeGUTS-package, 3
betacyfluthrinChronic, 4

concAC, 5
concAO, 5
concCst, 6
correlation_plot, 7
criteriaCheck, 7

dataGUTS, 8, *15*

fitBeeGUTS, 10
fitBetacyfluthrin_Chronic, 13

LCx, 14
LCx.beeSurvFit, 14, *14*

odeGUTS::predict_ode(), *21, 22*

plot.beeSurvData, 15
plot.beeSurvFit, 16
plot.beeSurvPred, 17
plot.beeSurvValidation, 18
plot.ppc, 19
ppc, 19
ppc.beeSurvFit, 20
ppc.beeSurvValidation, 20
predict.beeSurvFit, 21
predict2, 22
priorposterior_plot, 23

rstan::stan(), *11*

ShortTimeEffects, 24
ShortTimeEffects.beeSurvFit, *24*, 25
summary.beeSurvFit, 26
summary.LCx, 26

traceplot, 27
TRT, 28
TRT.beeSurvFit, 28, *28*

valBetacyfluthrinChronic, 29
validate, 30
validate.beeSurvFit, 31