Package ‘morseTKTD’

May 9, 2026

Title Bayesian Inference of TKTD Models

Version 0.1.3

Description Advanced methods for a valuable quantitative environmental risk assessment using Bayesian inference of survival Data with toxicokinetics toxicodynamics (TKTD) models. Among others, it facilitates Bayesian inference of the general unified threshold model of survival (GUTS). See models description in Jager et al. (2011) <doi:10.1021/es103092a> and implementation using Bayesian inference in Baudrot and Charles (2019) <doi:10.1038/s41598-019-47698-0>.

License AGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

Depends R (>= 3.5.0)

Imports deSolve, ggplot2, gridExtra, methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), rstantools (>= 2.4.0), testthat, zoo

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)

Suggests knitr, rmarkdown, xml2

Config/testthat/edition 3

VignetteBuilder knitr

SystemRequirements GNU make

NeedsCompilation yes

Author Virgile Baudrot [aut, cre],
Sandrine Charles [aut],
Marie Laure Delignette-Muller [aut],
Benoit Goussen [ctb],
Nils Kehrein [ctb],
Guillaume Kon-Kam-King [ctb],

Christelle Lopes [ctb],
Alexander Singer [ctb],
Philippe Veber [aut]

Maintainer Virgile Baudrot <`virgile.baudrot@qonfluens.com`>

Repository CRAN

Date/Publication 2025-06-04 15:00:02 UTC

Contents

`morseTKTD-package` *The 'morseTKTD' package.*

Description

Advanced methods for a valuable quantitative environmental risk assessment using Bayesian inference of survival Data with toxicokinetics toxicodynamics (TKTD) models. Among others, it facilitates Bayesian inference of the general unified threshold model of survival (GUTS).

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.32.7. https://mc-stan.org

---

`build_dN` *build sub-dataframe of Nsurv without NA*

Description

internal function to remove NA in 'replicate', 'time' and 'Nsurv' columns and building 'Nprec' variable. Removing 'conc" column.

Usage

```
build_dN(subdata)
```

Arguments

`subdata`	a list of data.frame

`build_dX` *build sub-dataframe of concentration without NA*

Description

internal function to remove NA in 'replicate', 'time' and 'conc' columns, remove 'Nsurv' column to only have concentration matrix

Usage

```
build_dX(subdata)
```

Arguments

`subdata`	a list of data.frame

`cadmium1` *Reproduction and survival data sets of chronic laboratory toxicity tests of cadmium with Daphnia*

Description

- **cadmium1**: Reproduction and survival data sets of chronic laboratory toxicity tests with *Daphnia magna* freshwater invertebrate exposed to five concentrations of cadmium during 21 days. Five concentrations were tested, with four replicates per concentration. Each replicate contained 10 organisms. Reproduction and survival were monitored at 10 time points.

Usage

```
data(cadmium1)
```

References

Billoir, E., Delhaye, H., Forfait, C., Clement, B., Triffault-Bouchet, G., Charles, S. and Delignette-Muller, M.L. (2012) Comparison of toxicity tests with different exposure time patterns: The added value of dynamic modelling in predictive ecotoxicology, *Ecotoxicology and Environmental Safety*, 75, 80-86.

`cadmium2` *Reproduction and survival data sets of chronic laboratory toxicity tests of cadmium with snails*

Description

- **cadmium2**: Reproduction and survival data sets of chronic laboratory toxicity tests with snails (*Lymnaea stagnalis*) exposed to six concentrations of cadmium during 28 days. Six concentrations were tested, with six replicates per concentration. Each replicate contained five organisms. Reproduction and survival were monitored at 17 time points.

Usage

```
data(cadmium2)
```

References

Ducrot, V., Askem, C., Azam, D., Brettschneider, D., Brown, R., Charles, S., Coke, M., Collinet, M., Delignette-Muller, M.L., Forfait-Dubuc, C., Holbech, H., Hutchinson, T., Jach, A., Kinnberg, K.L., Lacoste, C., Le Page, G., Matthiessen, P., Oehlmann, J., Rice, L., Roberts, E., Ruppert, K., Davis, J.E., Veauvy, C., Weltje, L., Wortham, R. and Lagadic, L. (2014) Development and validation of an OECD reproductive toxicity test guideline with the pond snail Lymnaea stagnalis (Mollusca, Gastropoda), *Regulatory Toxicology and Pharmacology*, 70(3), 605-14.

Charles, S., Ducrot, V., Azam, D., Benstead, R., Brettschneider, D., De Schamphelaere, K., Filipe Goncalves, S., Green, J.W., Holbech, H., Hutchinson, T.H., Faber, D., Laranjeiro, F., Matthiessen, P., Norrgren, L., Oehlmann, J., Reategui-Zirena, E., Seeland-Fremer, A., Teigeler, M., Thome, J.P., Tobor Kaplon, M., Weltje, L., Lagadic, L. (2016) Optimizing the design of a reproduction toxicity test with the pond snail Lymnaea stagnalis, *Regulatory Toxicology and Pharmacology*, vol. 81 pp.47-56.

`CheckData` *Set of function to test conformity of data*

Description

- **check_time**: check if the `time` within a `time serie` is (1) numeric, (2) unique, (3) minimal value is 0.
- **check_concentration**: check if the `concentration` is numeric and always positive.
- **check_Nsurv**: check if the `Nsurv` is (1) integer and (2) always positive (3) can be NA.
- **check_TimeNsurv**: check if the pair `time` - `Nsurv` within a `time serie` satisfies (1) Nsurv at t=0 is >0, (2) decreasing.
- **check_concNsurv**: check if the pair `conc` - `Nsurv` within a `time serie` satisfies that the timeline of concentration covers timeline of Nsurv.
- **checking_table**: add `msg` in a data.frame `data` if `check` are not all TRUE.
- **is_exposure_constant**: Test in a well-formed argument to function `SurvData` if the concentration is constant and different from `NA` for each replicate (each time-serie).
- **is.between**: Test if `x` is between `min` and `max`

Usage

```
check_time(data)

check_concentration(data)

check_Nsurv(data)

check_TimeNsurv(data)

check_concNsurv(data)

checking_table(data, check, msg)

is_exposure_constant(data)

is.between(x, min, max)
```

Arguments

`data`	a data.frame
`check`	binary vector of TRUE/FALSE
`msg`	a message to add to the data.frame
`x`	parameter to check if it's between min and max
`min`	minimal value. x must be greater than min
`max`	maximal value. x must be lower than max

Value

a boolean `TRUE` if concentration in `replicate` is constant, or `FALSE` if the concentration in at least one of the replicates is time-variable, and/or if `NA` occurs.

`chlordan` *Reproduction and survival data sets of chronic laboratory toxicity tests of chlordan with Daphnia*

Description

- **chlordan**: Reproduction and survival data sets of chronic laboratory toxicity tests with *Daphnia magna* freshwater invertebrate exposed to six concentrations of one organochlorine insecticide (chlordan) during 21 days. Six concentrations were tested, with 10 replicates per concentration. Each replicate contained one organism. Reproduction and survival were monitored at 22 time points. See Manar et al. (2009).

Usage

```
data(chlordan)
```

References

Manar, R., Bessi, H. and Vasseur, P. (2009) Reproductive effects and bioaccumulation of chlordan in Daphnia magna, *Environmental Toxicology and Chemistry*, 28(10), 2150-2159.

`ComputePredictSurvFit` *Compute post value on object*

Description

`compute_Nsurv`: compute the number of survival `Nsurv`

Usage

```
compute_Nsurv(x, ...)

compute_Ninit(x, ...)

## S3 method for class 'SurvPredict'
compute_Nsurv(x, Ninit = NULL, ...)
```

Arguments

`x`	an object of class `SurvPredict`
`...`	Further arguments to be passed to generic methods
`Ninit`	initial number of individual. Default is NULL.

Value

No return value, called for side effects. Return the same object after computing Number of survivor (`Nsurv` column) and number of initial individuals (`Ninit` column).

`copper` *Reproduction and survival data sets of chronic laboratory toxicity tests of copper on Daphnia*

Description

- **copper**: Reproduction and survival data sets of chronic laboratory toxicity tests with *Daphnia magna* freshwater invertebrate exposed to five concentrations of copper during 21 days. Five concentrations were tested, with three replicates per concentration. Each replicate contained 20 organisms. Reproduction and survival were monitored at 16 time points.

Usage

```
data(copper)
```

References

Billoir, E., Delignette-Muller, M.L., Pery, A.R.R. and Charles, S. (2008) A Bayesian Approach to Analyzing Ecotoxicological Data, *Environmental Science & Technology*, 42 (23), 8978-8984.

`dichromate` *Survival data set of chronic laboratory toxicity tests of dichromate with Daphnia*

Description

- **dichromate**: Survival data set of chronic laboratory toxicity tests with *Daphnia magna* freshwater invertebrate exposed to six concentrations of one oxidizing agent (potassium dichromate) during 21 days. Six concentrations were tested with one replicate of 50 organisms per concentration. Survival is monitored at 10 time points.

Usage

```
data(dichromate)
```

References

Bedaux, J., Kooijman, SALM (1994) Statistical analysis of toxicity tests, based on hazard modeling, *Environmental and Ecological Statistics*, 1, 303-314.

`Extract` *Extraction methods to recover output of fit object.*

Description

- **extract_Nsurv_ppc**: extract the `Nsurv` generated with the sampler. To be used for the Posterior Predictive Check (PPC).
- **extract_Nsurv_sim**: extract the `Nsurv` generated with the sampler. To be used for the Simulation (sim).
- **extract_param**: extract parameters of `SD` or `IT` models.
- **priors_distribution**: Return a `data.frame` with prior density distributions of parameters used in the model.

Usage

```
extract_Nsurv_ppc(fit)

extract_Nsurv_sim(fit)

extract_param(fit)

priors_distribution(fit, ...)

## S3 method for class 'SurvFit'
priors_distribution(fit, size_sample = 1000, ...)
```

Arguments

`fit`	An object of class `SurvFit`
`...`	Further arguments to be passed to generic methods
`size_sample`	Size of the random generation of the distribution.

Value

a `data.frame` with the extracted object from stanfit.

`FOCUSprofile` *A simulated exposure profile with 11641 time points using FOCUS model.*

Description

- **FOCUSprofile**: A simulated exposure profile with 11641 time points. Exposure profile of 11641 time points used for prediction. A data frame with 11641 observations on the following two variables: `time`, a vector of class `numeric`, `conc`, a vector of class `numeric` with exposure concentrations, and `replicate`, a vector of class `factor`.

Usage

```
data(FOCUSprofile)
```

`group_array` *build array with indices of rows*

Description

internal function to build an array from a data-frame to have the indices of rows

Usage

```
group_array(d)
```

Arguments

`d`	a data.frame

`LCxt` *Predict Lethal Concentration at any specified time point.*

Description

Predict the Lethal Concentration at any specified time point for a `SurvFit` object.

The function `LCx`, $x\backslash$ the dose required to kill $x\backslash$ after a specified test duration (`time_LCx`) (default is the maximum time point of the experiment).

Mathematical definition of $x\backslash$ denoted $LC(x,t)$, is:

$S(LC(x,t),t) = S(0,t) * (1 - x/100)$,

where $S(LC(x,t),t)$ is the survival probability at concentration $LC(x,t)$ at time t, and $S(0,t)$ is the survival probability at no concentration (i.e. concentration is 0) at time t which reflect the background mortality h_b:

$S(0,t) = exp(-hb * t)$.

In the function `LCx`, we use the median of $S(0,t)$ to rescale the $x\backslash$

Usage

```
lcxt(fit, x, t, ...)

## S3 method for class 'SurvFit'
lcxt(
  fit,
  x = 0.5,
  t = NULL,
  exposure_range = NULL,
  interpolate_length = 50,
  ...
)
```

Arguments

`fit`	An object used to select a method
`x`	rate of individuals dying (e.g., 0.5 for LC_{50}, 0.1 for LC_{10}, ...).
`t`	A number giving the time at which LC_x has to be estimated. If NULL, the latest time point of the experiment is used.
`...`	Further arguments to be passed to generic methods
`exposure_range`	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 experiment.
`interpolate_length`	of time point in the range of concentration between 0 and the maximal concentration. 100 by default. description.

Value

The function returns an object of class `LCx`, which is a list with the following information:

- X_propSurvival probability of individuals surviving considering the median of the background mortality (i.e. $S(0,t)*(1-x/100)$).
- X_prop_providedSurvival probability of individuals surviving as provided in arguments (i.e. $(100-X)/100$).
- time_LCxA number giving the time at which LC_x has to be estimated as provided in arguments or if NULL, the latest time point of the experiment is used.
- df_LCxA `data.frame` with quantiles (median, 2.5\ of LC_X at time `time_LCx` for X\
- df_doseA `data.frame` with four columns: `concentration`, and median `q50` and 95\ (`qinf95` and `qsup95`) of the survival probability at time `time_LCx`.

`LPxt` *Lethal Profile calculation*

Description

Predict the Lethal Profile factor leading to \$x\$ % of reduction in survival at a specific time \$t\$.

Generic method for `LPxt`, a function denoted $LP(x,t)$ for x\

Usage

```
lpxt(fit, x, ...)

## S3 method for class 'SurvFit'
lpxt(
  fit,
  x = 0.5,
  t = NULL,
  display.exposure = NULL,
  interpolate_length = NULL,
  max.steps = 100,
  accuracy = 0.01,
  ...
)

## S3 method for class 'LPxt'
update(object, accuracy = 0.01, max.steps = 100, ...)
```

Arguments

`fit`	An object used to select a method
`x`	rate of individuals dying (e.g., 0.5 for LP_{50}, 0.1 for LP_{10}, ...).
`...`	Further arguments to be passed to generic methods

`t`	A number giving the time at which LP_x has to be estimated. If NULL, the latest time point of the experiment is used.
`display.exposure`	A vector of the exposure porfile
`interpolate_length`	of time point in the range of concentration between 0 and the maximal concentration. 100 by default. description.
`max.steps`	max steps to find the LPxt
`accuracy`	accuracy of the LPxt algorithm (stop when reaching the accuracy).
`object`	An object of class `LPx`

Value

returns an object of class `LPxt`

`ModelData` *Create a list giving data to use in Bayesian inference.*

Description

Function to build the data list to give to stan

1. Order the data set in replicate and then in time to create a new column `i_row` used to delimited replicates.
2. Create a matrix of replicate and index "id_row"
3. Compute Nprec = lag of Nsurv
4. return a list of element to be passed to Stan sampler

Create a list of scalars giving priors to use in Bayesian inference.

Usage

```
modelData(data, model_type, ...)

## S3 method for class 'SurvData'
modelData(data, model_type = c(”SD”, ”IT”), hb_value = NULL, ...)

build_stanData(x)

build_priors(x, model_type = c(”SD”, ”IT”), hb_value = NULL)
```

Arguments

`data`	An object of class `SurvData`
`model_type`	TKTD model type ('SD' or 'IT')
`...`	Further arguments to be passed to generic methods
`hb_value`	default is NULL, can be fixed by specifying a numeric.
`x`	An object of class `survData`

Value

A list for parameterization of priors for Bayesian inference.

A list for parameterization of priors for Bayesian inference with JAGS.

`PlotData` *Plotting method for* `survDataVar` *objects*

Description

This is the generic `plot` S3 method for the `survDataVar` class. It plots the number of survivors as a function of time.

Usage

```
## S3 method for class 'SurvDataVarExp'
plot(
  x,
  xlab = "Time",
  ylab = "Number of survivors",
  main = NULL,
  one_plot = FALSE,
  add_legend = FALSE,
  ...
)

## S3 method for class 'SurvDataCstExp'
plot(
  x,
  xlab = "Time",
  ylab = "Number of survivors",
  main = NULL,
  one_plot = FALSE,
  ...
)
```

Arguments

`x`	an object of class `survData`.
`xlab`	a label for the X-axis, by default `Time`.
`ylab`	a label for the Y-axis, by default `Number of survivors`.
`main`	main title for the plot.
`one_plot`	if `TRUE`, draws all the points in one plot instead of one per `replicate` or `conc`.
`add_legend`	if `TRUE`, add the legend to the plot.
`...`	Further arguments to be passed to generic methods.

Value

an object of class `ggplot`, see function `ggplot`

`PlotLCxt` *Plot of the LCxt object*

Description

Method for plotting output of lcxt function returning object of class `LCxt`.

Usage

```
## S3 method for class 'LCxt'
plot(
  x,
  xlab = "Concentration",
  ylab = "Survival probability",
  main = NULL,
  ...
)
```

Arguments

`x`	an object of class `LCxt`
`xlab`	argument for the label of the x-axis
`ylab`	argument for the label of the y-axis
`main`	argument for the title of the graphic
`...`	Further arguments to be passed to generic methods

Value

an object of class `ggplot`, see function `ggplot`

`PlotLPxt` *Plot of the LPxt object*

Description

Method for plotting output of loxt function returning object of class `LPxt`.

Usage

```
## S3 method for class 'LPxt'
plot(
  x,
  plot = "curve",
  xlab = "Time",
  ylab = "Survival probability",
  main = NULL,
  ...
)
```

Arguments

`x`	an object of class `LPxt`
`plot`	style of the plot (default is curve)
`xlab`	argument for the label of the x-axis
`ylab`	argument for the label of the y-axis
`main`	argument for the title of the graphic
`...`	Further arguments to be passed to generic methods

Value

an object of class `ggplot`, see function `ggplot`

`PlotPPC` *Plot an object* `PPC`

Description

The coordinates of black points are the observed values of the number of survivors (pooled replicates) for a given concentration (X-axis) and the corresponding predicted values (Y-axis). 95\ value, colored in green if this interval contains the observed value and in red otherwise. The bisecting line (y = x) is added to the plot in order to see if each prediction interval contains each observed value. As replicates are shifted on the x-axis, this line is represented by steps.

Usage

```
## S3 method for class 'PPC'
plot(
  x,
  xlab = "Observation",
  ylab = "Prediction",
  main = NULL,
  dodge.width = 0,
  ...
)
```

Arguments

`x`	an object of class `PPC`
`xlab`	label of the x-axis
`ylab`	label of the y-axis
`main`	tital of the graphic
`dodge.width`	dodging width. Dodging preserves the vertical position of an geom while adjusting the horizontal position.
`...`	Further arguments to be passed to generic methods See `position_dodge` for further details.

Value

an object of class `ggplot`, see function `ggplot`

`PlotPredictSurvFit` *Plotting method for* `SurvPredict` *objects*

Description

This is the generic `plot` S3 method for the `SurvPredict` class. It plots concentration-response fit under target time survival analysis.

Usage

```
## S3 method for class 'SurvPredict'
plot(
  x,
  xlab = "Time",
  ylab = "Number of Survival",
  main = "Survival Probability with 95% Credible Interval",
  background_concentration = FALSE,
  add_legend = FALSE,
  ...
)
```

Arguments

`x`	an object of class `SurvPredict`
`xlab`	argument for the label of the x-axis
`ylab`	argument for the label of the y-axis
`main`	argument for the title of the graphic
`background_concentration`	Binary. If TRUE (default is FALSE), it print the background exposure profile.
`add_legend`	add legend to the plot, default is `FALSE`
`...`	Further arguments to be passed to generic methods

Value

an object of class `ggplot`, see function `ggplot`

`PlotPriorPosterior` *Plot of Prior and Posterior distributions*

Description

A function to plot priors and posteriors distribution after using the priorPosterior function on a SurvFit object.

Usage

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

Arguments

`x`	a `PriorPosterior` object
`...`	Further arguments to be passed to generic methods

Value

an object of class `ggplot`, see function `ggplot`

`PPC` *Posterior predictive check methods*

Description

This is the generic `ppc` S3 method for computing Posterior predictive check. It predicts values with 95 \ values for `SurvFit` objects.

Usage

```
ppc(fit, ...)

## S3 method for class 'SurvFit'
ppc(fit, ...)
```

Arguments

`fit`	An object of class `SurvFit`
`...`	Further arguments to be passed to generic methods

Value

a `data.frame` of class `PPC` with the original data point and the response of simulation and 95\ indicates if the observation fall within (green) or outside (red) of the 95\ credible interval.

`PredictSurvFit` *Prediction base on* `SurvFit` *objects*

Description

This is the generic `predict` S3 method for the `SurvFit` class. It provides predicted survival rate for "SD" or "IT" models under constant or time-variable exposure. prediction on constant exposure profile

Note: On constant exposure profiles, the results is explicit (exact), so you don't have to profile

Usage

```
predict_SurvFitCstExp(
  fit,
  display.exposure = NULL,
  hb_value = NULL,
  interpolate_length = NULL,
  ...
)
```

```
predict_cstSD(
  display.exposure = NULL,
  display.parameters = NULL,
  hb_value = NULL,
  interpolate_length = NULL
)

predict_cstIT(
  display.exposure = NULL,
  display.parameters = NULL,
  hb_value = NULL,
  interpolate_length = NULL
)

predict_SurvFitVarExp(
  fit,
  display.exposure = NULL,
  hb_value = NULL,
  interpolate_length = NULL,
  interpolate_method = "linear",
  ...
)

predict_varSD(
  display.exposure = NULL,
  display.parameters = NULL,
  hb_value = NULL,
  interpolate_length = NULL,
  interpolate_method = NULL
)

predict_varIT(
  display.exposure = NULL,
  display.parameters = NULL,
  hb_value = NULL,
  interpolate_length = NULL,
  interpolate_method = NULL
)

predict(fit, ...)

## S3 method for class 'SurvFit'
predict(
  fit,
  display.exposure = NULL,
  hb_value = NULL,
  interpolate_length = NULL,
  interpolate_method = "linear",
```

```
  ...
)
```

Arguments

`fit`	an object of class `SurvFit`
`display.exposure`	concentration points on which prediction is done
`hb_value`	a numeric used as `hb_value` (can be set to 0 to remove background mortality and take only effect parameters).
`interpolate_length`	if `display.time` is `NULL`, the argument `interpolate_length` can be used to provide a sequence from 0 to maximum of the time of exposure in original dataset (used for fitting).
`...`	Further arguments to be passed to generic methods
`display.parameters`	parameters of the specific model.
`interpolate_method`	The interpolation method for concentration. See package `deSolve` for details. Default is `linear`.

Value

a `list` of `data.frame` with the quantiles of outputs in `df_quantiles` or all the MCMC chains `df_spaghetti`

`PriorPosterior` *Return Prior and Posterior density of parameters of* `SurvFit` *object*

Description

This is the generic `pp` S3 method for the `survFitTT` class. It plots the predicted values with 95 \ values for `SurvFit` objects.

The coordinates of black points are the observed values of the number of survivors (pooled replicates) for a given concentration (X-axis) and the corresponding predicted values (Y-axis). 95\ value, colored in green if this interval contains the observed value and in red otherwise. The bisecting line (y = x) is added to the plot in order to see if each prediction interval contains each observed value. As replicates are shifted on the x-axis, this line is represented by steps.

Usage

```
priorPosterior(fit, ...)

## S3 method for class 'SurvFit'
priorPosterior(fit, ...)
```

Arguments

`fit`	An object of class `SurvFit`
`...`	Further arguments to be passed to generic methods

Value

a plot of class `ggplot`

`propiconazole` *Survival data set of chronic laboratory toxicity tests of propiconazole with Gammarus pulex*

Description

- **propiconazole**: Survival data set of chronic laboratory toxicity tests with *Gammarus pulex* freshwater invertebrate exposed to eight concentrations of one fungicide (propiconazole) during four days. Eight concentrations were tested with two replicates of 10 organisms per concentration. Survival is monitored at five time points.

Usage

```
data(propiconazole)
```

References

Nyman, A.-M., Schirmer, K., Ashauer, R., (2012) Toxicokinetic-toxicodynamic modelling of survival of *Gammarus pulex* in multiple pulse exposures to propiconazole: model assumptions, calibration data requirements and predictive power, *Ecotoxicology*, (21), 1828-1840.

`propiconazole_pulse_exposure` *Survival data set of chronic laboratory toxicity tests of propiconazole with Gammarus pulex*

Description

- **propiconazole_pulse_exposure**: Survival data set for *Gammarus pulex* exposed to propiconazole during 10 days with time-variable exposure concentration (non-standard pulsed toxicity experiments). Survival data set of laboratory toxicity tests with *Gammarus pulex* freshwater invertebrates exposed to several profiles of concentrations (time-variable concentration for each time series) of one fungicide (propiconazole) during 10 days.

Usage

```
data(propiconazole_pulse_exposure)
```

References

Nyman, A.-M., Schirmer, K., Ashauer, R., (2012) Toxicokinetic-toxicodynamic modelling of survival of *Gammarus pulex* in multiple pulse exposures to propiconazole: model assumptions, calibration data requirements and predictive power, *Ecotoxicology*, (21), 1828-1840.

`SurvData` *Creates a data set for survival analysis*

Description

This function creates a `SurvData` object from experimental data provided as a `data.frame`. The resulting object can then be used for plotting and model fitting. It can also be used to generate *individual-time* estimates.

The `x` argument describes experimental results from a survival toxicity test. Each line of the `data.frame` corresponds to one experimental measurement, that is a number of alive individuals at a given concentration at a given time point and in a given replicate. Note that either the concentration or the number of alive individuals may be missing. The data set is inferred to be under constant exposure if the concentration is constant for each replicate and systematically available. The function `survData` fails if `x` does not meet the expected requirements. Please run `survDataCheck` to ensure `x` is well-formed.

Usage

```
survData(data, ...)

## S3 method for class 'data.frame'
survData(data, ...)
```

Arguments

`data`	a `data.frame` containing the following four columns: • `replicate`: a vector of any class `numeric`, `character` or `factor` for replicate identification. A given replicate value should identify the same group of individuals followed in time • `conc`: a vector of class `numeric` with tested concentrations (positive values, may contain NAs) • `time`: a vector of class `integer` with time points, minimal value must be 0 • `Nsurv`: a vector of class `integer` providing the number of alive individuals at each time point for each concentration and each replicate (may contain NAs)
`...`	Further arguments to be passed to generic methods

Value

A dataframe of class `survData` and column `replicate` as `factor`.

See Also

`survDataCheck`

`SurvDataCheck` *Checks if an object can be used to perform survival analysis*

Description

The `survDataCheck` function can be used to check if an object containing survival data is formatted according to the expectations of the `survData` function.

Usage

```
survDataCheck(data, quiet = FALSE)
```

Arguments

`data`	any object looking as a data.frame.
`quiet`	binary (TRUE, FALSE). If FALSE (default), remove some messages in console.

Value

The function returns a dataframe with message describting the error in the formatting of the data. When no error is detected the object is empty.

`SurvFit` *Fits a TKTD model for survival analysis using Bayesian inference*

Description

This function estimates the parameters of a TKTD model ('SD' or 'IT') for survival analysis using Bayesian inference. In this model, the survival rate of individuals is modeled as a function of the chemical compound concentration with a mechanistic description of the effects on survival over time.

The function returns the parameter estimates of Toxicokinetic-toxicodynamic (TKTD) models `SD` for 'Stochastic Death' or `IT` fo 'Individual Tolerance'. TKTD models, and particularly the General Unified Threshold model of Survival (GUTS), provide a consistent process-based framework to analyse both time and concentration dependent datasets. In GUTS-SD, all organisms are assumed to have the same internal concentration threshold (denoted z), and, once exceeded, the instantaneous probability to die increases linearly with the internal concentration. In GUTS-IT, the threshold concentration is distributed among all the organisms, and once exceeded in one individual, this individual dies immediately.

This is the generic `plot` S3 method for the `SurvFit` class. It plots concentration-response fit under target time survival analysis.

Usage

```
fit(data, model_type, hb_value, ...)

## S3 method for class 'SurvDataCstExp'
fit(data, model_type, hb_value = NULL, ...)

## S3 method for class 'SurvDataVarExp'
fit(data, model_type, hb_value = NULL, ...)

## S3 method for class 'SurvFit'
plot(
  x,
  xlab = "Time",
  ylab = "Number of Survival",
  main = NULL,
  add_data = TRUE,
  add_legend = FALSE,
  ...
)
```

Arguments

`data`	An object of class `survDataCstExp` or `survDataVarExp`.
`model_type`	Can be `"SD"` or `"IT"` to choose between "Stochastic Death" or "Individual Tolerance" models (resp.). See the modeling vignette for details.
`hb_value`	If `TRUE`, the background mortality `hb` is taken into account. If `FALSE`, parameter `hb` is set to 0. The default is `TRUE`.
`...`	Further arguments to be passed to generic methods using argument of sampling function.
`x`	a `SurvFit` object
`xlab`	label of the x-axis, default is "Time",
`ylab`	label of the y-axis, default is "Number of Survival"
`main`	title of the plot, defaul is `NULL`,
`add_data`	to add original data to the plot. Default ir `TRUE`
`add_legend`	add legend to the plot, default is `FALSE`

Value

An object of class `stanfit` returned by `rstan::sampling`

an object of class `ggplot`, see function `ggplot`

References

Jager, T., Albert, C., Preuss, T. G. and Ashauer, R. (2011) General unified threshold model of survival-a toxicokinetic-toxicodynamic framework for ecotoxicology, *Environmental Science and Technology*, 45, 2529-2540. 303-314.

`survFit_TKTD_params` *Table of posterior estimated parameters*

Description

create the table of posterior estimated parameters for the survival analyses

Usage

```
survFit_TKTD_params(mcmc, model_type, hb_value = TRUE)
```

Arguments

`mcmc`	list of estimated parameters for the model with each item representing a chain
`model_type`	model type `SD` or `IT`
`hb_value`	TRUE or FALSE, conservning the use of hb parameter in the model.

Value

a `data.frame` with 3 columns (values, CIinf, CIsup) and 3-4rows (the estimated parameters)

`SurvIT_cst` *Internal predict function*

Description

Survival function for "IT" model with external concentration changing with time

Usage

```
SurvIT_cst(Cw, time, kd, hb, alpha, beta, interpolate_length = NULL)
```

Arguments

`Cw`	A vector of external concentration
`time`	A vector of time
`kd`	a vector of parameter
`hb`	a vector of parameter
`alpha`	a vector of parameter
`beta`	a vector of parameter
`interpolate_length`	can be used to provide a sequence from 0 to maximum of the time of exposure in original dataset (used for fitting).

Value

A data.frame with exposure columns `time` and `conc` and the resulting probabilisty of survival in `Psurv_XX` column where `XX` refer to an MCMC iteration

---

`SurvIT_var` *Internal predict function*

Description

Survival function for "IT" model with external concentration changing with time

Usage

```
SurvIT_var(
  Cw,
  time,
  kd,
  hb,
  alpha,
  beta,
  interpolate_length = NULL,
  interpolate_method = c("linear", "constant")
)
```

Arguments

`Cw`	A vector of external concentration
`time`	A vector of time
`kd`	a vector of parameter
`hb`	a vector of parameter
`alpha`	a vector of parameter
`beta`	a vector of parameter
`interpolate_length`	if `display.time` is `NULL`, the argument `interpolate_length` can be used to provide a sequence from 0 to maximum of the time of exposure in original dataset (used for fitting).
`interpolate_method`	The interpolation method for concentration. See package `deSolve` for details.

Value

A data.frame with exposure columns `time` and `conc` and the resulting probabilisty of survival in `Psurv_XX` column where `XX` refer to an MCMC iteration

---

`SurvSD_cst` *Internal predict function*

Description

Survival function for "SD" model with external concentration changing with time

Usage

```
SurvSD_cst(Cw, time, kd, hb, z, kk, interpolate_length = NULL)
```

Arguments

`Cw`	A vector of external concentration
`time`	A vector of time
`kd`	a vector of parameter
`hb`	a vector of parameter
`z`	a vector of parameter
`kk`	a vector of parameter
`interpolate_length`	can be used to provide a sequence from 0 to maximum of the time of exposure in original dataset (used for fitting).

Value

A data.frame with exposure columns `time` and `conc` and the resulting probabilisty of survival in `Psurv_XX` column where `XX` refer to an MCMC iteration

---

`SurvSD_var` *Internal predict function*

Description

Survival function for "SD" model with external concentration changing with time

Usage

```
SurvSD_var(
  Cw,
  time,
  kd,
  hb,
  z,
  kk,
  interpolate_length = NULL,
  interpolate_method = c("linear", "constant")
)
```

Arguments

`Cw`	A scalar of external concentration
`time`	A vector of time
`kd`	a vector of parameter
`hb`	a vector of parameter
`z`	a vector of parameter
`kk`	a vector of parameter
`interpolate_length`	if `display.time` is `NULL`, the argument `interpolate_length` can be used to provide a sequence from 0 to maximum of the time of exposure in original dataset (used for fitting).
`interpolate_method`	The interpolation method for concentration. See package `deSolve` for details.

Value

A data.frame with exposure columns `time` and `conc` and the resulting probabilisty of survival in `Psurv_XX` column where `XX` refer to an MCMC iteration

---

`zinc` *Reproduction and survival data sets of a chronic laboratory toxicity tests of zinc with Daphnia*

Description

- **zinc**: Reproduction and survival data sets of a chronic laboratory toxicity tests with *Daphnia magna* freshwater invertebrate exposed to four concentrations of zinc during 21 days. Four concentrations were tested with three replicates per concentration. Each replicate contained 20 organisms. Reproduction and survival were monitored at 15 time points.

Usage

```
data(zinc)
```

References

Billoir, E., Delignette-Muller, M.L., Pery, A.R.R. and Charles, S. (2008) A Bayesian Approach to Analyzing Ecotoxicological Data, *Environmental Science & Technology*, 42 (23), 8978-8984.

Index