Package 'drda'

June 10, 2021

Type Package

Title Dose-Response Data Analysis

Version 1.0.0

Description Fit logistic functions to observed dose-response data and evaluate goodness of fit measures. See Malyutina A., Tang J., and Pessia A. (2021) <doi:10.1101/2021.06.07.447323>.

License MIT + file LICENSE

URL https://github.com/albertopessia/drda

BugReports https://github.com/albertopessia/drda/issues

Encoding UTF-8

Suggests covr, knitr, testthat

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Alberto Pessia [aut, cre] (<https://orcid.org/0000-0001-8607-9191>), Alina Malyutina [ctb] (<https://orcid.org/0000-0001-8220-5859>)

Maintainer Alberto Pessia <dev@albertopessia.com>

Repository CRAN

Date/Publication 2021-06-10 07:50:07 UTC

R topics documented:

---

drda-package *Dose-response data analysis*

Description

drda is a package for fitting logistic curves and performing dose-response data analysis.

Available functions

Functions specific to drda:

- drda: main function for fitting observed data.
- logistic2_fn: 2-parameter logistic function.
- logistic4_fn: 4-parameter logistic function.
- logistic5_fn: 5-parameter logistic function.
- logistic6_fn: 6-parameter logistic function.
- gompertz_fn: Gompertz function.
- nauc: normalized area under the curve.
- naac: normalized area above the curve.

Functions expected for an object fit:

- anova: compare model fits.
- deviance: residual sum of squares of the model fit.
- logLik: value of the log-likelihood function associated to the model fit.
- plot: plotting function.
- predict: model predictions.
- print: basic model summaries.
- residuals: model residuals.
- sigma: residual standard deviation.
- summary: fit summaries.
- weights: model weights.

References

Malyutina A, Tang J, Pessia A (2021). drda: An R package for dose-response data analysis. bioRxiv, 2021.06.07.447323. doi: 10.1101/2021.06.07.447323

drda *Fit non-linear growth curves*

Description

Use the Newton's with a trust-region method to fit non-linear growth curves to observed data.

Usage

```
drda(
  formula,
  data,
  subset,
  weights,
  na.action,
  mean_function = "logistic4",
  is_log = TRUE,
  lower_bound = NULL,
  upper_bound = NULL,
  start = NULL,
  max_iter = 10000
)
```

Arguments

`formula`	an object of class `link[stats]{formula}` (or one that can be coerced to that class): a symbolic description of the model to be fitted. Currently supports only formulas of the type `y ~ x`.
`data`	an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `drda` is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process.
`weights`	an optional vector of weights to be used in the fitting process. If provided, weighted least squares is used with weights `weights` (that is, minimizing `sum(weights * residuals^2)`), otherwise ordinary least squares is used.
`na.action`	a function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The 'factory-fresh' default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
`mean_function`	the model to be fitted. See `details` for available models.
`is_log`	a logical value indicating whether the predictor variable `x` is already log-transformed. Default to `TRUE`. Set to `FALSE` if `x` is on its natural scale, i.e. strictly positive.

lower_bound — numeric vector with the minimum admissible values of the parameters. Use -Inf to specify an unbounded parameter.

upper_bound — numeric vector with the maximum admissible values of the parameters. Use Inf to specify an unbounded parameter.

start — starting values for the parameters.

max_iter — maximum number of iterations in the optimization algorithm.

Details

Available models:

Generalized logistic function:
The most general model in this package is the generalized logistic function selected by setting mean_function = "logistic6". It is defined in this package as the 6-parameter function
alpha + (beta - alpha) / (xi + nu * exp(-eta * (x - phi)))^(1 / nu)
where eta != 0, nu > 0, and xi > 0. Although beta can be any real value, we use the convention beta > alpha to avoid identifiability problems: when beta < alpha it is always possible to adjust the other parameters to obtain the same exact curve. When beta > alpha and eta > 0 the curve is monotonically increasing. If beta > alpha and eta < 0 the curve is monotonically decreasing.
Parameter alpha represents the lower horizontal asymptote of the curve. Parameter beta is related to the upper horizontal asymptote of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the value of the function at x = 0. Parameter nu affects near which asymptote maximum growth occurs. Parameter xi affects the value of the upper asymptote.
Note: the 6-parameter logistic function is non-identifiable from data and should not be used in real applications. It is available only for theoretical research convenience.

5-parameter logistic function:
The 5-parameter logistic function can be selected by choosing mean_function = "logistic5". The function is obtained by setting xi = 1 in the generalized logistic function, that is
alpha + (beta - alpha) / (1 + nu * exp(-eta * (x - phi)))^(1 / nu)
Parameter alpha represents the lower horizontal asymptote of the curve. Parameter beta represents the upper horizontal asymptote of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi is related to the value of the function at x = 0. Parameter nu affects near which asymptote maximum growth occurs.

4-parameter logistic function:
The 4-parameter logistic function is the default model of drda. It can be explicitly selected by choosing mean_function = "logistic4". The function is obtained by setting xi = 1 and nu = 1 in the generalized logistic function, that is
alpha + (beta - alpha) / (1 + exp(-eta * (x - phi)))
Parameter alpha represents the lower horizontal asymptote of the curve. Parameter beta represents the upper horizontal asymptote of the curve. Parameter eta represents the steepness (growth rate) of the curve. Parameter phi represents the x value at which the curve is equal to its mid-point, i.e. f(phi; alpha, beta, eta, phi) = (alpha + beta) / 2.

2-parameter logistic function:
The 2-parameter logistic function can be selected by choosing mean_function = "logistic2". The function is obtained by setting xi = 1, nu = 1, beta = 1, and alpha = 0 in the generalized logistic function, that is

`1 / (1 + exp(-eta * (x - phi)))`

Parameter `eta` represents the steepness (growth rate) of the curve. Parameter `phi` represents the `x` value at which the curve is equal to its mid-point, i.e. f(phi; eta, phi) = 1 / 2.

Gompertz function:

The Gompertz function is the limit for `nu -> 0` of the 5-parameter logistic function. It can be selected by choosing `mean_function = "gompertz"`. The function is defined in this package as

`alpha + (beta - alpha) * exp(-exp(-eta * (x - phi)))`

where `eta != 0`.

Parameter `alpha` represents the lower horizontal asymptote of the curve. Parameter `beta` represents the upper horizontal asymptote of the curve. Parameter `eta` represents the steepness (growth rate) of the curve. Parameter `phi` is related to the value of the function at `x = 0`.

Constrained optimization:

It is possible to search for the maximum likelihood estimates within pre-specified interval regions. Since the upper horizontal asymptote `beta` must be greater than the lower horizontal asymptote `alpha`, intervals are adjusted to satisfy this constraint.

Note: Hypothesis testing is not available for constrained estimates because asymptotic approximations might not be valid

Value

An object of class `drda` and `model_fit`, where `model` is the chosen mean function. It is a list containing the following components:

converged boolean value assessing if the optimization algorithm converged or not.

iterations total number of iterations performed by the optimization algorithm

constrained boolean value set to `TRUE` if optimization was constrained.

estimated boolean vector indicating which parameters were estimated from the data.

coefficients maximum likelihood estimates of the model parameters.

rss minimum value (found) of the residual sum of squares.

df.residuals residual degrees of freedom.

fitted.values fitted mean values.

residuals residuals, that is response minus fitted values.

weights (only for weighted fits) the specified weights.

mean_function model that was used for fitting.

n effective sample size.

sigma corrected maximum likelihood estimate of the standard deviation.

loglik maximum value (found) of the log-likelihood function.

fisher.info observed Fisher information matrix evaluated at the maximum likelihood estimator.

vcov approximate variance-covariance matrix of the model parameters.

call the matched call.

terms the `terms` object used.

model the model frame used.

na.action (where relevant) information returned by `model.frame` on the special handling of `NA`s.

is_log boolean value. It is `TRUE` if the predictor variable was given on the log scale.

`gompertz_fn` *Gompertz function*

Description

Evaluate at a particular set of parameters the Gompertz function.

Usage

```
gompertz_fn(x, theta)
```

Arguments

`x`	numeric vector at which the Gompertz function is to be evaluated.
`theta`	numeric vector with the four parameters in the form `c(alpha,beta,eta,phi)`.

Details

The Gompertz function f(x; theta) is defined here as
alpha + (beta - alpha) exp(-exp(-eta * (x - phi)))

where `theta` = `c(alpha,beta,eta,phi)`, `alpha` is the lower horizontal asymptote, `beta` > `alpha` is the upper horizontal asymptote, `eta` is the steepness of the curve or growth rate, and `phi` related to the function value at `x = 0`.

Value

Numeric vector of the same length of `x` with the values of the Gompertz function.

`logistic2_fn` *2-parameter logistic function*

Description

Evaluate at a particular set of parameters the 2-parameter logistic function.

Usage

```
logistic2_fn(x, theta)
```

Arguments

`x`	numeric vector at which the logistic function is to be evaluated.
`theta`	numeric vector with the parameters in the form `c(eta,phi)`.

Details

The 2-parameter logistic function f(x; theta) is defined here as

1 / (1 + exp(-eta * (x -phi)))

where theta = c(eta,phi), eta is the steepness of the curve or growth rate (also known as the Hill coefficient), and phi is the value of x at which the curve is equal to its mid-point, i.e. 1 / 2.

Value

Numeric vector of the same length of x with the values of the logistic function.

---

logistic4_fn *4-parameter logistic function*

Description

Evaluate at a particular set of parameters the 4-parameter logistic function.

Usage

```
logistic4_fn(x, theta)
```

Arguments

x	numeric vector at which the logistic function is to be evaluated.
theta	numeric vector with the four parameters in the form c(alpha,beta,eta,phi).

Details

The 4-parameter logistic function f(x; theta) is defined here as

alpha + (beta -alpha) / (1 + exp(-eta * (x -phi)))

where theta = c(alpha,beta,eta,phi), alpha is the lower horizontal asymptote, beta > alpha is the upper horizontal asymptote, eta is the steepness of the curve or growth rate (also known as the Hill coefficient), and phi is the value of x at which the curve is equal to its mid-point.

Value

Numeric vector of the same length of x with the values of the logistic function.

`logistic5_fn` *5-parameter logistic function*

Description

Evaluate at a particular set of parameters the 5-parameter logistic function.

Usage

```
logistic5_fn(x, theta)
```

Arguments

`x`	numeric vector at which the logistic function is to be evaluated.
`theta`	numeric vector with the five parameters in the form `c(alpha,beta,eta,phi,nu)`.

Details

The 5-parameter logistic function f(x; theta) is defined here as

`alpha + (beta -alpha) / (1 + nu * exp(-eta * (x -phi)))^(1 / nu)`

where `theta = c(alpha,beta,eta,phi,nu)`, `beta > alpha`, and `nu > 0`.

Value

Numeric vector of the same length of `x` with the values of the logistic function.

`logistic6_fn` *6-parameter logistic function*

Description

Evaluate at a particular set of parameters the 6-parameter logistic function.

Usage

```
logistic6_fn(x, theta)
```

Arguments

`x`	numeric vector at which the logistic function is to be evaluated.
`theta`	numeric vector with the six parameters in the form `c(alpha,beta,eta,phi,nu,xi)`.

Details

The 6-parameter logistic function f(x; theta) is defined here as

`alpha + (beta -alpha) / (xi + nu * exp(-eta * (x -phi)))^(1 / nu)`

where `theta = c(alpha,beta,eta,phi,nu,xi)`, `beta > alpha`, `xi > 0`, and `nu > 0`.

Value

Numeric vector of the same length of `x` with the values of the logistic function.

---

naac — *Area above the curve*

Description

Evaluate the normalized area above the curve (NAAC).

Usage

```
naac(object, xlim = c(-10, 10), ylim = c(0, 1))
```

Arguments

`object`	fit object as returned by `drda`.
`xlim`	numeric vector of length 2 with the lower and upped bound of the integration interval. Default is `c(-10,10)`.
`ylim`	numeric vector of length 2 with the lower and upped bound of the allowed function values. Default is `c(0,1)`.

Details

The area under the curve (AUC) is simply the integral of the chosen model y(x; theta) with respect to `x`.

In real applications the response variable is usually contained within a known interval. For example, if our response represents relative viability against a control compound, the curve is then expected to be between 0 and 1.

To make the AUC value comparable between different compounds and/or studies, this function set a hard constraint on both the `x` variable and the function `y`. The intervals can always be changed if needed.

The function f(x; theta) that is integrated here is defined as
f(x; theta) = ylim[1], if y(x; theta) < ylim[1]
f(x; theta) = y(x; theta), if ylim[1] <= y(x; theta) <= ylim[2]
f(x; theta) = ylim[2], if y(x; theta) > ylim[2]

Finally, the AUC is normalized by its maximum possible value, that is the area of the rectangle with width `xlim[2] -xlim[1]` and height `ylim[2] -ylim[1]`.

The normalized area above the curve is simply `NAAC = 1 -NAUC`.

Value

Numeric value representing the normalized area above the curve.

---

nauc — *Area under the curve*

Description

Evaluate the normalized area under the curve (NAUC).

Usage

```
nauc(object, xlim = c(-10, 10), ylim = c(0, 1))
```

Arguments

`object`	fit object as returned by `drda`.
`xlim`	numeric vector of length 2 with the lower and upped bound of the integration interval. Default is `c(-10,10)`.
`ylim`	numeric vector of length 2 with the lower and upped bound of the allowed function values. Default is `c(0,1)`.

Details

The area under the curve (AUC) is simply the integral of the chosen model y(x; theta) with respect to `x`.

In real applications the response variable is usually contained within a known interval. For example, if our response represents relative viability against a control compound, the curve is then expected to be between 0 and 1.

To make the AUC value comparable between different compounds and/or studies, this function set a hard constraint on both the `x` variable and the function `y`. The intervals can always be changed if needed.

The function f(x; theta) that is integrated here is defined as
f(x; theta) = ylim[1], if y(x; theta) < ylim[1]
f(x; theta) = y(x; theta), if ylim[1] <= y(x; theta) <= ylim[2]
f(x; theta) = ylim[2], if y(x; theta) > ylim[2]

Finally, the AUC is normalized by its maximum possible value, that is the area of the rectangle with width `xlim[2] -xlim[1]` and height `ylim[2] -ylim[1]`.

Value

Numeric value representing the normalized area under the curve.

`plot.drda` *Model fit plotting*

Description

Plot maximum likelihood curves fitted with `drda`.

Usage

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

Arguments

`x`	`drda` object as returned by the `link[drda]{drda}` function.
`...`	other `drda` objects or parameters to be passed to the plotting functions. See 'Details'.

Details

This function provides a scatter plot of the observed data, overlaid with the maximum likelihood curve fit. If multiple fit objects are given, they will all be placed in the same plot.

Accepted plotting arguments are:

base character string with the base used to print the values on the x axis. Accepted values are `e` for the natural logarithm (the default), `10` for base 10, and `2` for base 2.

col curve color(s). By default, up to 9 color-blind friendly colors are provided.

xlab, ylab axis labels.

xlim, ylim the range of x and y values with sensible defaults.

level level of confidence intervals. Set to zero or a negative value to disable confidence intervals.

legend custom labels for the legend model names.

Value

No return value.

Index