Package ‘basicdrm’

April 26, 2024

Title Fit Hill Dose Response Models

Version 0.3.0

Description Evaluate, fit, and analyze Hill dose response models (Goutelle et al., 2008 <doi:10.1111/j.1472-8206.2008.00633.x>), also sometimes referred to as four-parameter log-logistic models. Includes tools to invert Hill models, select models based on the Akaike information criterion (Akaike, 1974 <doi:10.1109/TAC.1974.1100705>) or Bayesian information criterion (Schwarz, 1978 <`https://www.jstor.org/stable/2958889`>), and construct bootstrapped confidence intervals both on the Hill model parameters and values derived from the Hill model parameters.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10), stats

NeedsCompilation no

Author Anang Shelat [aut],
Nathaniel Twarog [aut, cre]

Maintainer Nathaniel Twarog <`nathaniel.twarog@stjude.org`>

Repository CRAN

Date/Publication 2024-04-26 16:20:02 UTC

R topics documented:

calcHillBootstrap *Estimate Bootstrapped Confidence Intervals on Hill Model Parameters*

Description

By bootsttrapping a large number of vectors, this function estimates confidence intervals on the paramters of the given Hill model. If the model already has confidence intervals estimated, they will be replaced with a warning

Usage

```
calcHillBootstrap(hfit, ciLevs = c(0.025, 0.975), numBoot = NULL)
```

Arguments

`hfit`	An object of class `hillrm`
`ciLevs`	The lower and upper p-values for the estimated confidence interval. The default values, 0.025 and 0.975, produce a 95% confidence interval.
`numBoot`	The number of bootstrapped coefficients to be sampled. If `NULL` (the default) will be selected to have at least 10 samples lie the selected interval, with a minimum of 100 and a maximum of 1000.

Value

An object of class `hillrm`, containing all the values found in any `hillrm` object (see `fitHillModel()`) as well as the following three values:

- `ciLevs`: the values used to set the bounds of the confidence intervals
- `ciCoefs`: a width-4 array of all bootstrapped Hill model coefficents sampled by the function
- `ciMat`: a 2-by-4 array of values representing the estimated confidence intervals on the four Hill model parameters

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)

hfit <- fitHillModel(conc,response,c(1,2,3,4),start=c(0.5,1,0,100))
cihfit <- calcHillBootstrap(hfit)
```

`calcHillConfInt` *Estimate an confidence interval on a Hill model property*

Description

Given a function from Hill model parameters to one or more model properties, this function produces a confidence interval on that value or those values using the bootstrapped model coefficents produced by `calcHillBootstrap()`. This is useful for estimating confidence intervals on other values like IC50, or generating confidence intervals on fitted values for plots.

Usage

```
calcHillConfInt(hfit, parfunc, civals = NULL)
```

Arguments

`hfit` An object of class `hillrm`, with the `ciCoefs` property produced by `calcHillBootstrap()`

`parfunc` A function from a four parameter Hill model vector (see `evalHillModel()`) to a single value or a vector of values with a consisten length

`civals` An optional set of upper and lower bounds on the confidence interval to be estimated. If `NULL`, the default, the `ciLevs` property from `calcHillBootstrap()` will be used.

Value

An n-by-3 array, where n is the length of the vector produced by `parfunc`. The first row is the lower bound of the confidence interval, the second row is the function evaluated at the best-fit Hill model, and the third row is the upper bound of the confidence interval.

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)

hfit <- fitHillModel(conc,response,c(1,2,3,4),start=c(0.5,1,0,100))
cihfit <- calcHillBootstrap(hfit)

ic50_ci <- calcHillConfInt(cihfit,function(h) invertHillModel(50,h))
```

calculateHillAUC *Calculate the area under the curve for a Hill dose response model*

Description

The area under the curve, or AUC, is a commonly used and robust metric for evaluating and comparing dose response models. The area is calculated in a log-concentration space, and so is dependent not only on the concentration bounds, but also on the base of the logarithm used. By default, this function follows the common convention of using base 10.

Usage

```
calculateHillAUC(hpar, range, baseline = 0, logbase = 10)
```

Arguments

`hpar`	A four parameter vector specifying the Hill model. Parameter details are found in the documentation for `evalHillModel()`
`range`	A two element vector specifying the lower and upper bound of area being calculated
`baseline`	The reference baseline response around which the area is being calculated. The default value of 0 is generally the most intuitive choice, but a value of 1 (or 100 in percent) could be used if the dose-response model is fitting relative survival.
`logbase`	The base of the logarithm applied to concentrations

Value

A single value specifying the area under the curve in the given range

Examples

```
auc <- calculateHillAUC(c(1,3,0,75), c(0.05,10))

# Area *over* the curve in survival studies
aoc <- -calculateHillAUC(c(0.1,2,1,0.1), c(0.01,1), baseline=1)
```

estimateAIC *Estimate Akaike Information Criterion*

Description

Estimate Akaike Information Criterion

Usage

```
estimateAIC(resid, npar)
```

Arguments

`resid`	A vector of residuals from a given fit
`npar`	The number of paramters in the model

Value

The Akaike informtion criterion (AIC) value for the fit

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)

hfit4p <- fitHillModel(conc,response,c(1,2,3,4),start=c(0.5,1,0,100))
hfit3p <- fitHillModel(conc,response,c(1,2,4),start=c(0.5,1,0,100))

aic4p <- estimateAIC(residuals(hfit4p),4)
aic3p <- estimateAIC(residuals(hfit3p),3)
```

`estimateBIC` *Estimate Bayesian Information Criterion*

Description

Estimate Bayesian Information Criterion

Usage

```
estimateBIC(resid, npar)
```

Arguments

`resid`	A vector of residuals from a given fit
`npar`	The number of paramters in the model

Value

The Bayesian informtion criterion (BIC) value for the fit

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)

hfit4p <- fitHillModel(conc,response,c(1,2,3,4),start=c(0.5,1,0,100))
hfit3p <- fitHillModel(conc,response,c(1,2,4),start=c(0.5,1,0,100))
```

```
aic4p <- estimateBIC(residuals(hfit4p),4)
aic3p <- estimateBIC(residuals(hfit3p),3)
```

evalHillModel *Evaluate a Hill dose response model*

Description

Evaluate a Hill dose response model

Usage

```
evalHillModel(conc, hpar)
```

Arguments

conc	A vector of concentrations (including 0 or Inf)
hpar	A four parameter vector specifying the Hill model. The values of the parameter vector are, in order, the dose of median effect (also often referred to as the EC50), the Hill slope, the minimal effect (observed when no drug or dose is present), and the maximal effect (theoretically observed when the drug or dose is infinite).

Value

A vector of response values the same length as `conc`

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,100)

response <- evalHillModel(conc, hpar)
```

findBestHillModel *Selects a best-fitting Hill model given defaults*

Description

Using the function `fitHillModel()`, this function fits four Hill models with minimal and maximal effects either varying or fixed at the given default values; it then selects the best fitting model based on the Bayesian information criterio or Akaike information criterion, and returns a Hill fit object with information from all fits included.

Usage

```
findBestHillModel(
  formula,
  data,
  defaults,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  useBIC = TRUE
)
```

Arguments

`formula` Either an object of class `formula` such as would be provided to a modeling function like `lm()`, or a numeric vector of concentration values (including 0 or Inf)

`data` If `forumula` is a symbolic formula, a data frame containing the specified values. If `formula` is a numeric vector of concentrations, a numeric vector of response values

`defaults` A two value numeric vector containing the default minimal effect and the default maximal effect, in that order

`weights` A vector of weights (between 0 and 1) the same length as `conc` and `act` which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If `NULL` (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in `data` if `formula` is a symbolic formula

`start` A vector of four starting values for the Hill model to be fit. Any values not being fit will be fixed at these starting values. If left as `NULL`, a starting vector will be estimated from the data.

`direction` Determines the possible directionality of the dose response model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effect is *greater* than the minimal effect. If less than 0, the fitting wll require tha the maximal effect is *less* than the minimal effect.

`lower` A length-four vector of lower bounds on the Hill parameter values. Any parameters for which you do not wish to specify a bound can be set to `NA`.

`upper` A vector of upper bounds on the Hill parameter values. Works the same as parameter `lower`.

`useBIC` Determines the information criterion to be used. If `TRUE` (the default), uses the Bayesian information criterion. If `FALSE`, uses the Akaike information criterion

Value

An object of class `hillrm`. Contains all of the values found in any `hillrm` object (see `fitHillModel()`), as well as `allfits`, a named list of lists containing the `coefficients` and `parvectors` for each of the individual fits, as well as the Bayesian information criterion (`bic`) and Akaike informtion criterion (`aic`) values for each fit.

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)

hfit <- findBestHillModel(conc,response,defaults=c(0,100))
```

`fitHillModel` *Fit a Hill dose response model to data*

Description

This function uses the R `stats` function `optim` to fit a Hill dose response model to a given set of dose and response values. Four different model settings are allowed, in which the minimal and maximal effects are either fixed at a provided value or allowed to be fit to the data.

Usage

```
fitHillModel(
  formula,
  data,
  model,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)
```

Arguments

`formula` Either an object of class `formula` such as would be provided to a modeling function like `lm()`, or a numeric vector of concentration values (including 0 or Inf)

`data` If `forumula` is a symbolic formula, a data frame containing the specified values. If `formula` is a numeric vector of concentrations, a numeric vector of response values

`model` A vector of values between 1 and 4, specifying the precise model to be fit. The values correspond to the four parameters of the Hill model: dose of median effect, Hill slope, minimal effect, and maximal effect (see `evalHillModel()`). The first of these two are always fit, so `model` must contain at least `1` and `2`. The presence of `3` or `4` will determine if those parameters are also fit, or fixed at the given starting value. So `c(1,2,4)` will fit the dose of median effect, the

Hill slope, and the maximal effect, but will leave the minimal effect fixed at the starting value.

`weights` A vector of weights (between 0 and 1) the same length as `conc` and `act` which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If `NULL` (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in `data` if `formula` is a symbolic formula

`start` A vector of four starting values for the Hill model to be fit. Any values not being fit will be fixed at these starting values. If left as `NULL`, a starting vector will be estimated from the data, but it will almost always be better to provide an explicit staring model.

`direction` Determines the possible directionality of the dose response model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effect is *greater* than the minimal effect. If less than 0, the fitting wll require tha the maximal effect is *less* than the minimal effect.

`lower` A vector of lower bounds on the Hill parameter values. Can be the same length as `model` (in which case the bounds will be applied to the corresponding fit parameters) or the full length of 4. Any parameters for which you do not wish to specify a bound can be set to `NA`.

`upper` A vector of upper bounds on the Hill parameter values. Works the same as parameter `lower`.

Value

An object of class `hillrm`, containing the following values:

- `conc`: the given vector of concentraitons
- `act`: the given vector of responses
- `weights`: the vector of measurement weights used in minimizing the sum of squared errors
- `coefficients`: the full four-parameter Hill parameter vector (accessible by the function `coef()`)
- `par`: the vector of paramters that were actually fit
- `fitted.values`: the predicted responses of the best fit model (accessible by the functoin `fitted()`)
- `residuals`: the difference between the actual responses and the predicted responses (accessible by the function `residuals()`)
- `model`: the vector of values between 1 and 4 specifying the precise model that was fit
- `mname`: a character string naming the precise model that was fit. One of "m2p", "m3plc", "m3puc", or "m4p"
- `start`: a four-value parameter vector used as the starting value for the model fit
- `direction`: the direction constraint used in the fit
- `pbounds`: a two-by-four matrix of values specifying the lower and upper bounds used in the fit

Examples

```
conc <- c(0,2^(-6:3),Inf)
hpar <- c(1,3,0,75)
response <- evalHillModel(conc, hpar) + rnorm(length(conc),sd=7.5)
data <- data.frame(conc=conc,response=response,weight=c(0.5,rep(1,10),0.1))

hfit <- fitHillModel(conc,response,c(1,2,3,4),start=c(0.5,1,0,100))
hfit2 <- fitHillModel(response~conc,data,c(1,2,4),weight,start=c(0.5,1,0,100),
                      direction=0,lower=c(NA,NA,0,0))
```

`invertHillModel` *Calculates the required doses of a Hill dose response model*

Description

This funciton takes one or more desired response values and determines the doses that will produce the desired effects given a particular Hill dose response model. This is useful for estimating things like IC50.

Usage

```
invertHillModel(effect, hpar, invalidNA = FALSE)
```

Arguments

`effect`	A vector of desired response values
`hpar`	A four parameter vector specifying the Hill model. Parameter details are found in the documentation for `evalHillModel()`
`invalidNA`	Specifies what to do with values that are outside the range of the given Hill model. If `FALSE` (the default), values "below" the given range will be set to zero, and values "above" the given range will be set to Inf. If `TRUE`, invalid values will be set to `NA`.

Value

A vector of concentrations the same length as `effect`.

Examples

```
invertHillModel(0.5, c(1,2,0,0.7))

invertHillModel(seq(0.1,0.9,by=0.1), c(0.1,4,0,0.65), invalidNA=TRUE)
```

Index