

The backfitRichards Package

March 28, 2008

Type Package

Title Backfitted independent values of Richards curves

Version 0.4

Date 2008-03-27

Author Jens Henrik Badsberg

Address Statens Serum Institut, Denmark

Maintainer Jens Henrik Badsberg <march2008spam4richards@badsberg.eu>

Depends R (>= 2.0.0), richards (>= 0.1)

Description Backfitted independent values of Richards curves are computed and plotted

License GPL (>= 2)

R topics documented:

backConf	2
backFit	3
backFits	4
backPlot	4
backfitRichards-package	5
fits2backFits	7

Index	9
--------------	----------

backConf

*Pointwise confidence intervals for 'backPlot'***Description**

Pointwise confidence intervals for 'backPlot'.

Usage

```
backConf(Z, x = as.real(dimnames(Z)[[1]]), ylim = c(min(Z), max(Z)),
  log = "", lineLty = 2, lineCol = "black", lineLwd = 0.5,
  centerLty = 2, centerCol = "blue", centerLwd = 1,
  confCex = 1, confCol = "blue", confLwd = 1.5, ...)
```

Arguments

Z	a matrix with dependent values.
ylim	a numeric vector with the range of the y axis.
x	a numeric vector with the independent values.
log	the argument log to matplot .
lineLty	a numeric value, lty for the line y = 100 %.
lineCol	a text string, col for the line y = 100 %.
lineLwd	a numeric value, lwd for the line y = 100 %.
centerLty	a numeric value, lty for joining the means of the dependent values.
centerCol	a text string, col for joining the means of the dependent values.
centerLwd	a numeric value, lwd for joining the means of the dependent values.
confCex	a numeric value, cex for the size of the 'square' of the confidence intervals.
confCol	a text string, col of the confidence intervals.
confLwd	a numeric value, lwd of the confidence intervals.
...	additional arguments.

Details

Pointwise confidence intervals for 'backPlot'.

Author(s)

Jens Henrik Badsberg

backFit

*Backfitted values for one Richards curve***Description**

Calculate the backfitted values for one Richards curve.

Usage

```
backFit(X, FUN = NULL, fit = NULL, ny = 0,
        respName = "SIGNAL", indepName = "ARGX",
        applyFUN2X = function(X, FUN, ...) FUN(X, ...), ...)
```

Arguments

X	the dataset for FUN.
FUN	a function estimating a model for each dataset. This function is only called if the argument <code>fit</code> is not given.
fit	a optional fit.
ny	a (optional) numerical value. This value has to given, if <code>ny</code> is not fitted in the model.
respName	a text string with the name of the dependent variable.
indepName	a text string with the name of the independent variable..
applyFUN2X	a function applied to each dataset, retrieving starting value and fixing outliers, and then calling the function <code>FUN</code> to estimate the model for the modified dataset. This function is only called if the argument <code>fit</code> is not given.
...	additional arguments.

Details

The backfitted values for one Richards curve are calculated: For each value of the independent variable (`indepName`) the observed mean of the dependent variable (`respName`) is found from the dataset `X`. The value is found of the independent variable for which the value of the estimated Richards function is equal to the observed mean of the response, by the inverse of the Richards function and the fit of the curve. This estimated value of the independent variable is taken relative the actual value of the independent variable.

Value

Matrix with relevant values.

Author(s)

Jens Henrik Badsberg

backFits *Backfitted values for a list of Richards curves*

Description

Backfitted values are calculated, with fit of the curves, for a list of Richards curves.

Usage

```
backFits(list, FUN, ...)
```

Arguments

list	a list of datasets.
FUN	a function estimating a model for each dataset.
...	additional arguments.

Details

Calculates the backfitted values for a list of Richards curves, with computing the fits of the curves.

Value

List of matrices with relevant values, see [backFit](#).

Author(s)

Jens Henrik Badsberg

backPlot *Plot backfitted values for Richards curves*

Description

Backfitted values for Richards curves are plotted.

Usage

```
backPlot(backFits = NULL, xt = function(x) 1/x, ylim = c(min(Z), max(Z)),
  Conf = backConf, main = "Relative backfitted concentration",
  sub = paste("Above / below / NA: ",
    paste("", c(length(which(Z > ylim[2])),
      length(which(Z < ylim[1])),
      length(which(is.na(Z))))),
    collapse = " /")),
  xlab = "Concentration [ Dilution step ]",
  ylab = "Recovery (%)", ...)
```

Arguments

backFits	a list of backFits, see backFit .
ylim	a numeric vector with the range of the y axis.
xt	a function for transforming the x axis.
Conf	a function for adding confidence intervals, default backConf .
main	an overall title for the plot: see title .
sub	a sub title for the plot: see title .
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
...	additional arguments, to matplot and <code>Conf</code> , see matplot and backConf .

Details

backfitted values for Richards curves are plottet.

Value

Matrix with the plottet values.

Author(s)

Jens Henrik Badsberg

backfitRichards-package

Backfitted independent values of Richards curves

Description

Backfitted values of the independent variable of Richards curves are computed and plotted.

Details

Package:	backfitRichards
Type:	Package
Version:	0.4
Date:	2008-03-27
License:	GPL Version 2 or later

Author(s)

Jens Henrik Badsberg

Maintainer: Jens Henrik Badsberg <march2008spam4richards@badsberg.eu>

References

Richards, F.J.: A flexible growth function for empirical use. *J. Exp. Bot.* 1959; 10:290-300.
 Finney DJ.: *Statistical methods in biological assay*, 3rd ed. London: Griffin; 1978.
 Finney DJ.: Response curves for radioimmunoassay. *Clinical Chemistry* 1983; 29:1562-1566.
 Schofield T.: Assay Validation. In: *Encyclopedia of Biopharmaceutical Statistics*. New York: Marcel Dekker 2000; p. 21-30.
<http://en.wikipedia.org/wiki/ELISA>

Examples

```
## Not run:

library(nlme)

data(ChickWeight)

ChickWeight <- cbind(ChickWeight, ExpTime = exp(ChickWeight[, "Time"]))

groupedChicks <- groupedData(weight ~ ExpTime | Diet / Chick,
                             inner = ~ PR, data = ChickWeight)

listChicks <- split(groupedChicks, list(groupedChicks[, "Diet"],
                                       groupedChicks[, "Chick"]),
                   drop = TRUE)

fit.Fpl.BP <- function(X, start, lower, upper)
{
  try(nls(weight ~ SSny1(ExpTime, a, d, b, x50),
          lower = lower, upper = upper, algorithm = "port",
          control = nls.control(warnOnly = TRUE), data = X))
}

Start <- list(a = 25, d = 200, b = 10, x50 = 10, ny = 1)
Lower <- list(a = -2000, d = 5, b = 0, x50 = 1, ny = 0.01)
Upper <- list(a = 50, d = 1500, b = 50, x50 = 10^20, ny = 128)

myApplyFUN2X <-
  function(X, FUN, start = Start, lower = Lower, upper = Upper,
           Report = print, ...) {
    if (dim(X)[1] > 0) {
      result <- FUN(X, start, lower, upper)
      if (is.null(result) | class(result) == "try-error")
        Report(X)
      return(result)
    }
  }
```

```

fits.Fpl.BP <- fitList(listChicks, FUN = fit.Fpl.BP, n = 4,
                      applyFUN2X = myApplyFUN2X)

back.Fits <- backFits(listChicks, FUN = fit.Fpl.BP, ny = 1,
                     respName = "weight", indepName = "ExpTime",
                     applyFUN2X = myApplyFUN2X)

par(mfrow = c(2, 2))

Z.back <- backPlot(backFits = back.Fits, xt = function(x) x,
                  main = "Relative backfitted exp(time)", xlab = "Exp(Time)",
                  ylim = c(0, 500), log = "x", pch = 1, col = "red")

Z.back <- backPlot(backFits = back.Fits, xt = function(x) log(x),
                  main = "Relative backfitted exp(time)", xlab = "Time",
                  ylim = c(0, 500), pch = 1, col = "red")

Z.back <- backPlot(backFits = back.Fits, xt = function(x) log(x),
                  main = "Relative backfitted exp(time)", xlab = "Time",
                  log = "y", ylim = c(1, 5000), pch = 1, col = "red")

Z.back <- backPlot(backFits = back.Fits, xt = function(x) log(x),
                  main = "Relative backfitted exp(time)", xlab = "Time",
                  log = "y", ylim = c(10, 500), pch = 1, col = "red")

## End(Not run)

```

fits2backFits

Backfitted values from fits of Richards curves

Description

Compute backfitted values from the data and the fits of Richards curves.

Usage

```
fits2backFits(list, fits, ...)
```

Arguments

<code>list</code>	list of datasets.
<code>fits</code>	list of fits.
<code>...</code>	additional arguments.

Details

Compute backfitted values from the data and the fits of Richards curves, see [backFit](#). The function [backFits](#) does not have the argument `fit`, and has to fit the models. This function `fits2backFits` takes the fits as argument. The backfitted values are the found by applying [backFit](#) to the i -te element of `list` and the i -te element of `fit`. Thus the two list has to match.

Value

List of matrices with relevant values.

Author(s)

Jens Henrik Badsberg

Index

*Topic **package**

- backConf, [1](#)
- backFit, [2](#)
- backfitRichards-package, [5](#)
- backFits, [3](#)
- backPlot, [4](#)
- fits2backFits, [7](#)

backConf, [1](#), [4](#)
backFit, [2](#), [4](#), [7](#)
backfitRichards
 (*backfitRichards-package*),
 [5](#)

backfitRichards-package, [5](#)
backFits, [3](#), [7](#)
backPlot, [4](#)

fits2backFits, [7](#)

matplot, [2](#), [4](#)

title, [4](#)