

Package ‘grofit’

February 20, 2015

Type Package

Title The package was developed to fit fit many growth curves obtained under different conditions.

Version 1.1.1-1

Date 2012-08-05

Author Matthias Kahm, Maik Kschischo

Maintainer Billy Aung Myint <r@use-r.com>

Description The package was developed to fit fit many growth curves obtained under different conditions in order to derive a conclusive dose-response curve, for instance for a compound that potentially affects growth. grofit fits data to different parametric models (function gcFitModel) and in addition provides a model free spline fit (function gcFitSpline) to circumvent systematic errors that might occur within application of parametric methods.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2014-10-06 15:21:13

NeedsCompilation no

R topics documented:

grofit-package	2
drBootSpline	3
drFit	5
drFitSpline	6
gcBootSpline	8
gcFit	9
gcFitModel	11
gcFitSpline	12
gompertz	13

gompertz.exp	14
grofit	15
grofit.control	16
grofit.data	18
grofit.time	18
initgompertz	19
initgompertz.exp	19
initlogistic	20
inrichards	21
logistic	22
low.integrate	23
plot.drBootSpline	23
plot.drFit	24
plot.drFitSpline	25
plot.gcBootSpline	26
plot.gcFit	27
plot.gcFitModel	28
plot.gcFitSpline	29
ran.data	31
richards	32
summary.drBootSpline	33
summary.drFit	33
summary.drFitSpline	34
summary.gcBootSpline	35
summary.gcFit	35
summary.gcFitModel	36
summary.gcFitSpline	37
Index	38

grofit-package	<i>The package was developed to fit many growth curves obtained under different conditions.</i>
----------------	---

Description

The package was developed to fit many growth curves obtained under different conditions in order to derive a conclusive dose-response curve, for instance for a compound that potentially affects growth. grofit fits data to different parametric models (function gcFitModel) and in addition provides a model free spline fit (function gcFitSpline) to circumvent systematic errors that might occur within application of parametric methods. This amendment increases the reliability of the characteristic parameters (e.g. lag phase, maximal growth rate, stationary phase) derived from a single growth curve. By relating obtained parameters to the respective condition (e.g. concentration of a compound) a dose response curve can be derived (function drFitSpline) that enables the calculation of descriptive pharma-/toxicological values like half maximum effective concentration (EC50). Bootstrap and cross-validation techniques are used for estimating confidence intervals of all derived parameters (functions gcBootSpline, drBootSpline).

Details

Package: grofit
 Type: Package
 Version: 1.0
 Date: 2009-12-08
 License: GPL-2
 LazyLoad: yes

The main functions are

gcFitModel	Fits certain growth laws to growth data.
gcFitSpline	Fits a smoothed spline to growth data.
gcBootSpline	Generates a bootstrap sample of fitted splines (growth curves).
drFitModel	Fits a smoothed spline to dose-response data.
drBootSpline	Generates a bootstrap sample of fitted splines (dose-response curves).

Author(s)

Matthias Kahm, Maik Kschischo

Maintainer: Matthias Kahm <sysbio_software@rheinahr-campus.de>

References

C. Birch, (1999), A new Generalized Logistic Sigmoid Growth Equation Compared with the Richards Growth Equation, *Annals of Botany*, p. 713-723, vol. 83.

G. Hasenbrink, L. Kolacna, J. Ludwig, H. Sychrova, M. Kschischo and H. Lichtenberg-Frate, (2006), Ring Test Assessment of the mKir2.1 Growth Based Assay in *Saccharomyces cerevisiae* Using Parametric Models and Model-free Fits, *Applied and Microbiology Biotechnology*, p. 1212-1221, vol. 73.

M. H. Zwietering, I. Jongenburger, F. M. Rombouts and K. van 'T Riet, (1990), Modeling of the Bacterial Growth Curve, *Applied and Environmental Microbiology*, p. 1875-1881, vol. 56.

drBootSpline	<i>Function to create a bootstrap sample of splines fitted to dose-response curves.</i>
--------------	---

Description

The function passes the numeric vectors `conc` and `test` to the function `drFitSpline`, which fits a smoothed spline to the data and estimates the EC50. Calling `drFitSpline` several times (specified by the options set in `control`) creates a bootstrap sample in form of a list of objects of class `drFitSpline`.

Usage

```
drBootSpline(conc, test, drID = "undefined",
             control = grofit.control())
```

Arguments

conc	Numeric vector, containing concentration information for dose-response. curve fit and EC50 estimation.
test	Numeric vector, containing response values related to concentrations.
drID	Character, identifying the dose-response data.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function <code>grofit.control</code> .

Value

Generates an object of class `drBootSpline`

raw.conc	Raw data given to the function; equivalent to <code>conc</code> .
raw.test	Raw data given to the function; equivalent to <code>test</code> .
drID	String identifier given to the function; equivalent to <code>drID</code> .
boot.conc	Table of concentration values per column, resulting from each spline fit of the bootstrap.
boot.test	Table of response values per column, resulting from each spline fit of the bootstrap.
boot.drSpline	List containing all <code>drFitSpline</code> objects generated by the call of <code>drFitSpline</code> .
ec50.boot	Vector of estimated EC50 values from each bootstrap entry.
bootFlag	Logical, indicating an empty bootstrap sample.
control	Object of class <code>grofit.control</code> containing list of options passed to the function as <code>control</code> .

See Also

[drFitSpline](#), [summary.drBootSpline](#), [plot.drBootSpline](#)

Examples

```
x <- 1:50
y <- 30/(1+exp(-0.5*(25-x)))+rnorm(50)
TestRun <- drBootSpline(x,y,"ID",grofit.control(nboot.dr=50))
print(summary(TestRun))
plot(TestRun)
```

drFit	<i>Function to run the dose-response curve fit and EC50 estimation</i>
-------	--

Description

The function serves to estimate several dose-response curves at once. The function calls the functions [drFitSpline](#) and [drBootSpline](#) to generate a table with estimates for EC50 and respecting statistics.

Usage

```
drFit(gcFitData, control = grofit.control())
```

Arguments

gcFitData	Data frame containing the data for the dose-response curve estimation. It must follow the convention: 1. column: string, naming the experiment for which a dose-response curve shall be estimated; 2. column: string, additional information about the experiment; 3. column: numeric, concentration of substrate; 4. column: logical, reliability flag, 5. column: string, naming a parametric model, 6. column: logical, indicating wether a logarithmic transformation was applied to the time points of the growth data; 7. column: logical, indicating wether a logarithmic transformation was applied to the growth values of the growth data; 8.-55. characteristic growth parameters obtained from growth curve fit.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Details

The function is meant to process the result from the fit of several growth curves performed by [gcFit](#). It therefore expects a certain input format. For each of the distinct experiments specified by the names in the first column, the function relates the concentration (3. column) to a characteristic growth parameter (8.-55. columns). The choice of the parameter depends on the option parameter specified by the [grofit.control](#) function. The concentration and parameter are passed to [drFitSpline](#) and [drBootSpline](#), which perform the dose-response curve estimation and bootstrapping.

Value

Generates an object of class `drFit`

raw.data	Data that passed to the function as <code>gcFitData</code> .
drTable	Data frame, 1. column: string, naming the experiment; 2. column: logical, indicating wether a logarithmic transformation was applied to concentration data; 3. column, logical, indicating wether a logarithmic transformation was applied to growth parameter; 4. column, numeric, number of bootstrap samples used; 5.-8. column: numeric, estimated EC50 value and respecting reponse value (5.-6.),

also in original scale if a transformation is applied (7.-8.); 9.-10. column: numeric, mean and standard dev. from bootstrap sample; 11.-14. column: 90 and 95 percent confidence interval from bootstrap sample; 15. column: mean EC50 of bootstrap sample in original scale; 16.-19. 90 and 95 percent confidence interval in original scale.

drBootSplines	List of all drBootSpline objects generated by the call of <code>drBootSpline</code> for each distinct experiments. Note: access to each object via double brace: <code>LIST[[1]]</code> , <code>LIST[[2]]</code> ...
drFittedSplines	list of all drFitSpline objects generated by the call of <code>drFitSpline</code> for each distinct experiments. Note: access to each object via double brace: <code>LIST[[1]]</code> , <code>LIST[[2]]</code> ...
control	Object of class <code>grofit.control</code> containing list of options passed to the function as <code>control</code> .

See Also

[drFitSpline](#), [drBootSpline](#), [drBootSpline](#), [summary.drFit](#), [plot.drFit](#)

Examples

```
# generate random growth curve data set
foo <- ran.data(100, 25)
time <- foo$time
data <- foo$data
# fit dose response curves
drData <- gcFit(time, data, control = grofit.control(interactive = FALSE))
# use the output of gcFit as an input for drFit
drFit.result <- drFit(summary(drData))
print(summary(drFit.result))
plot(drFit.result)
```

drFitSpline

Function to fit smoothed splines to dose reponse data.

Description

The function fits dose response curves.

Usage

```
drFitSpline(conc, test, drID = "undefined",
            control = grofit.control())
```

Arguments

conc	Numeric vector, concentration (dose) data.
test	Numeric vector, response data belonging to conc.
drID	Character, identifying the dose response data.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Details

The function uses the R internal function [smooth.spline](#) to fit a spline to the provided data. From the resulting curve the EC50 value is calculated.

Value

Generates an object of class `drFit`

raw.conc	Raw data provided to the function as <code>conc</code> .
raw.test	Raw data provided to the function as <code>test</code> .
drID	Character, identifying the dose response data.
fit.conc	Fitted concentration values.
fit.test	Fitted response values.
spline	nls object generated by the smooth.spline function.
fitFlag	Logical, indicating whether a spline could be fitted successfully to data.
reliable	Logical, indicating whether the provided data is reliable (to be set manually).
control	Object of class <code>grofit.control</code> containing a list of options passed to the function as <code>control</code> .
parameters	List of parameters estimated from dose response curve fit.
EC50	Half maximal concentration.
yEC50	Response value related to EC50.
EC50.orig	EC50 value in original scale, if a transformation was applied.
xEC50.orig	Response value for EC50 in original scale, if a transformation was applied.

See Also

[drFit](#), [summary.drFitSpline](#), [plot.drFitSpline](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(-0.5*(15-x)))+rnorm(30)/20
TestRun <- drFitSpline(x,y)
print(summary(TestRun))
plot(TestRun)
```

gcBootSpline	<i>Function to generate a bootstrap sample from a dataset of growth data.</i>
--------------	---

Description

Bootstrapping of a growth curve.

Usage

```
gcBootSpline(time, data, gcID = "undefined",
              control = grofit.control())
```

Arguments

time	Numeric vector containing the data for x-axes.
data	Numeric vector giving the growth values belonging to each element of time.
gcID	Vector (of any length) identifying the growth curve data.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Details

A bootstrap sample (with replacement) is generated from the original data. For each of the bootstrap samples, the characteristic parameters λ , μ , A and integral are estimated.

Value

Generates an object of class `gcBootSpline`

raw.time	Raw data given to the function; equivalent to <code>time</code> .
raw.data	Raw data given to the function; equivalent to <code>data</code> .
gcID	Identifier, given to the function as <code>gcID</code> .
boot.time	Table of concentration values per column, resulting from each spline fit of the bootstrap.
boot.data	Table of response values per column, resulting from each spline fit of the bootstrap.
boot.gcSpline	List containing all <code>gcFitSpline</code> objects generated by the call of gcFitSpline .
lambda	Vector containing all the estimates from the bootstrap sample for the lag-phase.
mu	Vector containing all the estimates from the bootstrap sample for the maximum slope.
A	Vector containing all the estimates from the bootstrap sample for the maximum growth value.
integral	Vector containing all the estimates from the bootstrap sample for the integral.

bootFlag	Logical, indicating an empty bootstrap sample.
control	Object of class <code>grofit.control</code> containing list of options passed to the function as control.

See Also

[gcFitModel](#), [gcFitSpline](#), [gcFit](#), [summary.gcBootSpline](#), [plot.gcBootSpline](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(0.5*(15-x)))+rnorm(30)/20
TestRun <- gcBootSpline(x,y,"ID",grofit.control(nboot.gc=50))
print(summary(TestRun))
plot(TestRun)
```

gcFit	<i>Function to run growth curve fitting on several data sets containing growth data.</i>
-------	--

Description

The function serves to estimate several growth curves at once. The function calls the functions [gcFitSpline](#), [gcFitModel](#) and [gcBootSpline](#) to generate estimates for growth parameters and respecting statistics.

Usage

```
gcFit(time, data, control = grofit.control())
```

Arguments

time	Numeric matrix, consisting of n columns for the time points and m rows for the different experiments.
data	Data frame consisting of 3+n columns, containing three columns of additional information and n columns of growth data corresponding to time. Format convention: 1. column, character as an experiment identifier; 2. column: character, additional information about respecting experiment; 3. column: concentration of substrate of a compound under which the experiment is obtained; 4.-(n+3). column: growth data corresponding to the time points in time.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Value

Generates an object of class `gcFit`

<code>raw.time</code>	Raw time passed to the function as <code>time</code> .
<code>raw.data</code>	Raw time passed to the function as <code>data</code> .
<code>gcTable</code>	Table with characteristic growth values and related statistics related for each growth curve data provided to the function. This table, which is also returned by the generic <code>summary.gcFit</code> method applied to an <code>gcFit</code> object, is used as an input for <code>drFit</code> . 1. column: character, naming the experiment; 2. column: logical, indicating whether a logarithmic transformation was applied to concentration data; 3. column, logical, indicating whether a logarithmic transformation was applied to growth parameter; 4. column, numeric, number of bootstrap samples used; 5.-8. column: numeric, estimated EC50 value and respecting response value (5.-6.), also in original scale if a transformation is applied (7.-8.); 9.-10. column: numeric, mean and standard dev. from bootstrap sample; 11.-14. column: 90 and 95 percent confidence interval from bootstrap sample; 15. column: mean EC50 of bootstrap sample in original scale; 16.-19. 90 and 95 percent confidence interval in original scale.
<code>gcFittedModels</code>	All <code>gcFitModel</code> objects, generated by the call of <code>gcFitModel</code> . Note: access to each object via double brace: <code>gcFittedModels[[#n]]</code> .
<code>gcFittedSplines</code>	All <code>gcFitSpline</code> objects, generated by the call of <code>gcFitSpline</code> . Note: access to each object via double brace: <code>gcFittedSplines[[#n]]</code> .
<code>gcBootSplines</code>	All <code>gcFitSpline</code> objects, generated by the call of <code>gcFitSpline</code> . Note: access to each object via double brace: <code>gcFittedSplines[[#n]]</code> .
<code>control</code>	Object of class <code>grofit.control</code> containing list of options passed to the function as <code>control</code> .

See Also

[grofit](#), [grofit.time](#), [grofit.data](#), [ran.data](#), [summary.gcFit](#), [plot.gcFit](#)

Examples

```
# generate random growth curve data sets,
# fulfilling the conventions described in arguments
foo <- ran.data(100, 25)
time <- foo$time
data <- foo$data
# fit dose response curves
result <- gcFit(time,data, control=grofit.control(fit.opt="b"))
print(summary(result))
plot(result)
```

gcFitModel	<i>Function to fit parametric models to growth data.</i>
------------	--

Description

Performs a parametric fit of certain models. The model with the best AIC is returned.

Usage

```
gcFitModel(time, data, gcID = "undefined",
           control = grofit.control())
```

Arguments

time	Numeric vector containing the data for x-axes.
data	Numeric vector giving the growth values belonging to each element of time.
gcID	Vector (of any length) identifying the growth curve data.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Details

The function calls [gcFitSpline](#) that uses the R internal function [lowess](#) to estimate initial values for the parametric fit. Four different models were tested by default. By means of an AIC criterion it is decided which model fits the data best. The tested models are: Gompertz law, modified Gompertz law, logistic law and Richards law.

Note: If a certain model depicts not an appropriate description of a given data set [nls](#) might stop and produce an error. This error stems from the generation of infinite or missing values or singular gradients in the optimization algorithm. These errors are not to be taken critical and indicates only that a certain model is not an appropriate description of a certain growth curve. When trying a couple of models it is usual that some of them can not be fitted. To prevent an overflow of almost redundant error messages ("nls(formulamodel, start = init.model) : singular gradient", "Error in numericDeriv(form[[3L]], names(ind), env) : missing value or an infinite produced by the mo they are displayed only as short error message on screen. If errors were frequently produced in models the user expects to be suitable, a change of the initial value definition (see e.g. [initgompertz](#), [initlogistic](#)) might help.

Value

Generates an object of class `gcFitModel`

raw.time	Raw data given to the function; equivalent to time.
raw.data	Raw data given to the function; equivalent to data.
gcID	Identifier, given to the function as gcID.
fit.time	Vector of fitted concentration values.

fit.data	Vector of fitted growth values.
parameters	List of estimated growth values.
A	Maximum growth value.
mu	Maximum slope.
lambda	Lag-phase.
integral	Integral under growth curve.
model	String naming the parametric model used.
nls	nls object generated by the R internal function nls .
reliable	Logical, indicating wether the provided data is reliable (to be set manually).
fitFlag	Logical, indicating wether a model could fitted successfully to data.
control	Object of class <code>grofit.control</code> containing list of options passed to the function as <code>control</code> .

See Also

[gompertz](#), [gompertz.exp](#), [richards](#), [logistic](#), [gcFitSpline](#), [summary.gcFitModel](#), [plot.gcFitModel](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(0.5*(15-x)))+rnorm(30)/20
TestRun <- gcFitModel(x,y)
print(summary(TestRun))
plot(TestRun)
```

gcFitSpline

Function to fit smoothed splines to growth data.

Description

The function fits a smoothed spline to a set of growth data.

Usage

```
gcFitSpline(time, data, gcID = "undefined",
            control = grofit.control())
```

Arguments

time	Numeric vector containing the data for x-axes.
data	Numeric vector giving the growth values belonging to each element of time.
gcID	Vector (of any length) identifying the growth curve data.
control	Object of class <code>grofit.control</code> containing a list of options generated by the function grofit.control .

Value

raw.time	Raw data given to the function; equivalent to time.
raw.data	Raw data given to the function; equivalent to data.
gcID	Vector (of any length) identifying the growth curve data.
fit.time	Vector of fitted concentration values.
fit.data	Vector of fitted growth values.
parameters	List of estimated growth values from spline fit: A, maximum growth value; mu, maximum slope; lambda, lag-phase, integral, integral under growth curve.
parametersLowess	List of estimated growth values from lowess fit: A, maximum growth value; mu, maximum slope; lambda, lag-phase.
spline	smooth.spline object generated by smooth.spline .
reliable	Logical, indicating whether the provided data is reliable (to be set manually).
fitFlag	Logical, indicating whether a spline could be fitted successfully to data.
control	Object of class <code>gprofit.control</code> containing list of options passed to the function as <code>control</code> .

See Also

[gcFitModel](#), [gcBootSpline](#), [summary.gcFitSpline](#), [plot.gcFitSpline](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(0.5*(15-x)))+rnorm(30)/20
TestRun <- gcFitSpline(x,y)
print(summary(TestRun))
plot(TestRun)
```

gompertz

Gompertz growth law

Description

The function calculates the values of the Gompertz growth curve for given time points.

Usage

```
gompertz(time, A, mu, lambda, addpar = NULL)
```

Arguments

time	Time points (x-axes) for which the function values will be returned.
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.
addpar	Additional parameters have no effect in this type of model. They belong to the standard model description in <code>grofit</code> and are initialized as <code>addpar=NULL</code> in the function header.

Details

The function follows the parametrization

$$y(t) = A \cdot \exp \left[- \exp \left(\frac{\mu \cdot \exp(1)}{A} (\lambda - t) + 1 \right) \right]$$

Value

Numeric vector giving the values of Gompertz growth law for numeric input vector `time`.

Examples

```
x <- 1:30
y <- gompertz(x, 10, 2, 5)
plot(x,y)
```

gompertz.exp

Modified Gompertz growth law

Description

The function calculates the values of the modified Gompertz growth curve for given time points.

Usage

```
gompertz.exp(time, A, mu, lambda, addpar)
```

Arguments

time	Time points (x-axes) for which the function values will be returned
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.
addpar	Numeric vector of size two, <code>addpar[1]</code> corresponds to scaling parameter α and <code>addpar[2]</code> corresponds to shifting parameter t_{shift} .

Details

The function follows the parametrization

$$y(t) = A \cdot \exp \left[- \exp \left(\frac{\mu \cdot \exp(1)}{A} (\lambda - t) + 1 \right) \right] + A \cdot \exp(\alpha(t - t_{shift}))$$

Value

Numeric vector giving the values of modified Gompertz growth law for numeric input vector `time`.

Examples

```
x <- 1:30
y <- gompertz.exp(x, 10, 2, 5, c(1,2))
plot(x,y)
```

grofit

Standard workflow

Description

The function enables the user to run all features of the **grofit** package as a standard workflow. First several growth curves are fitted by usage of the function [gcFit](#). Afterwards respecting dose-response curves are estimated from by the function [drFit](#).

Usage

```
grofit(time, data, ec50 = FALSE, control = grofit.control())
```

Arguments

<code>time</code>	Numeric matrix, consisting of <code>n</code> columns for the time points and <code>m</code> rows for the different experiments.
<code>data</code>	Data frame consisting of <code>3+n</code> columns, containing three columns of additional information and <code>n</code> columns of growth data corresponding to <code>time</code> . 1. column, character as an experiment identifier; 2. column: character, additional information about respecting experiment; 3. column: concentration of substrate of a compound under which the experiment is obtained; 4.-(<code>n+3</code>). column: growth data corresponding to the time points in <code>time</code> .
<code>ec50</code>	Logical, indicating whether dose-response curves and EC50 should be estimated after the growth curve fit.
<code>control</code>	Object of class <code>grofit.control</code> containing a list of options specified by grofit.control .

Value

time	Numeric matrix, equivalent to input parameter time.
data	Data frame, equivalent to input parameter data.
gcFit	Object of class gcFit.
drFit	Object of class drFit.
control	Object of class grofit.control containing a list of options passed to the function as control.

See Also

[grofit.time](#), [grofit.data](#)

grofit.control	<i>Function to set all available grofit options</i>
----------------	---

Description

The function generates an object of class grofit.control with the different **grofit** options and also allows to change these options.

Usage

```
grofit.control(neg.nan.act = FALSE, clean.bootstrap = TRUE,
              suppress.messages = FALSE, fit.opt = "b",
              log.x.gc = FALSE, log.y.gc = FALSE,
              interactive = TRUE, nboot.gc = 0,
              smooth.gc= NULL, model.type=c("logistic",
              "richards", "gompertz", "gompertz.exp"),
              have.atleast = 6, parameter = 9, smooth.dr = NULL,
              log.x.dr = FALSE, log.y.dr = FALSE, nboot.dr = 0)
```

Arguments

neg.nan.act	Logical, indicates wether the program should stop when negative growth values or NA values appear (TRUE). Otherwise the program removes this values silently (FALSE). Improper values may be caused by incorrect data or input errors. Default: FALSE.
clean.bootstrap	Logical, determines if negative values which occur during bootstrap should be removed (TRUE) or kept (FALSE). Note: Infinite values are always removed. Default: TRUE.
suppress.messages	Logical, indicates wether grofit messages (information about current growth curve, EC50 values etc.) should be displayed (FALSE) or not (TRUE). This option is meant to speed up the processing of high throuput data. Note: warnings are still displayed. Default: FALSE.

<code>fit.opt</code>	Indicates whether the program should perform a model fit ("m"), a spline fit ("s") or both ("b"). Default: "b".
<code>log.x.gc</code>	Logical, indicates whether a $\ln(x + 1)$ should be applied to the time data of the growth curves. Default: FALSE.
<code>log.y.gc</code>	Logical, indicates whether a $\ln(y + 1)$ should be applied to the growth data of the growth curves. Default: FALSE.
<code>interactive</code>	Logical, controls whether the fit of each growth curve is controlled manually by the user. Default: TRUE.
<code>nboot.gc</code>	Number of bootstrap samples used for the model free growth curve fitting. Use <code>nboot.gc=0</code> to disable the bootstrap. Default: 0.
<code>smooth.gc</code>	Parameter describing the smoothness of the spline fit; usually (not necessary) in (0;1]. Set <code>smooth.gc=NULL</code> causes the program to query an optimal value via cross validation techniques. Note: This is partly experimental. In future improved implementations of the <code>smooth.spline</code> function may lead to different results. See documentation of the R function <code>smooth.spline</code> for further details. Especially for datasets with few data points the option NULL might result in a too small smoothing parameter, which produces an error in <code>smooth.spline</code> . In that case the usage of a fixed value is recommended. Default: NULL.
<code>model.type</code>	Character vector giving the names of the parametric models which should be fitted to the data. Default: <code>c("gompertz", "logistic", "gompertz.exp", "richards")</code> .
<code>have.atleast</code>	Minimum number of different values for the response parameter one should have for estimating a dose response curve. Note: All fit procedures require at least six unique values. Default: 6.
<code>parameter</code>	The column of the output table which should be used for creating a dose response curve. See documentation of <code>gcFit</code> , <code>drFit</code> or <code>summary.gcFit</code> for further details. Default: 9, which represents the maximum slope of the parametric growth curve fit.
<code>smooth.dr</code>	Smoothing parameter used in the spline fit by <code>smooth.spline</code> during dose response curve estimation. Usually (not necessary) in (0; 1]. See documentation of <code>smooth.spline</code> for further details. Default: NULL.
<code>log.x.dr</code>	Logical, indicates whether a $\ln(x + 1)$ should be applied to the concentration data of the dose response curves. Default: FALSE.
<code>log.y.dr</code>	Logical, indicates whether a $\ln(y + 1)$ should be applied to the response data of the dose response curves. Default: FALSE.
<code>nboot.dr</code>	Numeric value, defining the number of bootstrap samples for EC50 estimation. Use <code>nboot.dr=0</code> to disable bootstrapping. Default: 0.

Value

Generates a list containing the entries described above.

Examples

```
# default option
DefOpt <- grofit.control()
# user defined
MyOpt <- grofit.control(smooth.gc=0.5, model.type=c("gompertz", "logistic"))
```

`grofit.data`*Example of growth data*

Description

Data frame containing several datasets of growth curves. Format: 1. column: a character vector, naming each experiment which is performed for the different concentrations specified in the 3. column; 2. column: a character vector, giving additional information about the experiments; 3. column: a numeric vector, containing the concentration of a compound under which the different experiments are obtained; 4.-68. column: a numeric vectors, giving the growth values corresponding to the time points specified in [grofit.time](#).

Usage

```
data(grofit.data)
```

Format

A data frame with 7 observations on the following 68 variables. Format: 1. column: a character vector, naming each experiment which is performed for the different concentrations specified in the 3. column; 2. column: a character vector, giving additional information about the experiments; 3. column: a numeric vector, containing the concentration of a compound under which the different experiments are obtained; 4.-68. column: a numeric vectors, giving the growth values corresponding to the time points specified in [grofit.time](#).

See Also

[grofit.time](#)

`grofit.time`*Example of growth data.*

Description

Numeric matrix containing a several datasets of time points corresponding to [grofit.data](#).

Usage

```
data(grofit.time)
```

Format

A data frame with 7 observations on the following 65 variables. a numeric vectors giving time points corresponding to the growth values in [grofit.data](#).

See Also

[grofit.data](#)

initgompertz	<i>Function to generate initial values for parameter estimation</i>
--------------	---

Description

The function receives time points, growth data, values for A , μ and λ and returns a list object which entries are used as initial values in the nonlinear fit procedure [nls](#).

Usage

```
initgompertz(time, y, A, mu, lambda)
```

Arguments

time	Time points.
y	Growth data.
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.

Value

A	Maximum of the curve.
mu	Maximum slope.
lambda	Lag-phase.
addpar	additional parameters are not used in this type of model.

See Also

[gompertz](#), [gcFitModel](#)

initgompertz.exp	<i>Function to generate initial values for parameter estimation</i>
------------------	---

Description

The function receives time points, growth data, values for A , μ and λ and returns a list object which entries are used as initial values in the nonlinear fit procedure [nls](#). In this case the function is serve to define initial values for scaling parameter α and shifting parameter t_{shift}

Usage

```
initgompertz.exp(time, y, A, mu, lambda)
```

Arguments

time	Time points.
y	Growth data.
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.

Value

A	Maximum of the curve.
mu	Maximum slope.
lambda	Lag-phase.
addpar	Two element vector defining scaling parameter α and shifting parameter t_{shift} .

See Also

[gompertz.exp](#), [gcFitModel](#)

initlogistic

Function to generate initial values for parameter estimation

Description

The function receives time points, growth data, values for A , μ and λ and returns a list object which entries are used as initial values in the nonlinear fit procedure [nls](#).

Usage

```
initlogistic(time, y, A, mu, lambda)
```

Arguments

time	Time points.
y	Growth data.
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.

Value

A	Maximum of the curve.
mu	Maximum slope.
lambda	Lag-phase.
addpar	additional parameters are not used in this type of model.

See Also

[logistic](#), [gcFitModel](#)

initrichards

Function to generate initial values for parameter estimation

Description

The function receives time points, growth data, values for A , μ and λ and returns a list object which entries are used as initial values in the nonlinear fit procedure [nls](#). In this type of model initial values for shape exponent ν .

Usage

```
initrichards(time, y, A, mu, lambda)
```

Arguments

time	Time points.
y	Growth data.
A	Maximum of the curve. If a vector is provided only the first entry is used.
mu	Maximum slope. If a vector is provided only the first entry is used.
lambda	Lag-phase. If a vector is provided only the first entry is used.

Value

A	Maximum of the curve.
mu	Maximum slope.
lambda	Lag-phase.
addpar	Shape exponent ν .

See Also

[richards](#), [gcFitModel](#)

`logistic`*Logistic growth law*

Description

The function calculates the values of the logistic growth curve for given time points.

Usage

```
logistic(time, A, mu, lambda, addpar = NULL)
```

Arguments

<code>time</code>	Time points (x-axes) for which the function values will be returned.
<code>A</code>	Maximum of the curve. If a vector is provided only the first entry is used.
<code>mu</code>	Maximum slope. If a vector is provided only the first entry is used.
<code>lambda</code>	Lag-phase. If a vector is provided only the first entry is used.
<code>addpar</code>	Additional parameters have no effect in this type of model. They belong to the standard model description in <code>profit</code> and are initialized as <code>addpar=NULL</code> in the function header.

Details

The function follows the parametrization

$$y(t) = \frac{A}{1 + \exp\left(\frac{4\mu}{A}(\lambda - t) + 2\right)}$$

Value

Numeric vector giving the values of logistic growth law for numeric input vector `time`.

Examples

```
x <- 1:30
y <- logistic(x, 10, 2, 5)
plot(x,y)
```

low.integrate	<i>Function to estimate the area under a curve given as respecting x and y(x) values</i>
---------------	--

Description

Approximates the integral of function by calculating the area under a smoothed spline fitted to the given x and y(x) values.

Usage

```
low.integrate(x, y)
```

Arguments

x	Numeric vector.
y	Numeric vector, corresponding to x.

Details

The function uses the the R internal function [smooth.spline](#).

Value

Area under the smoothed spline.

See Also

[smooth.spline](#)

plot.drBootSpline	<i>Generic plot function for drBootSpline objects</i>
-------------------	---

Description

Generic plot function for drBootSpline objects. Plots the raw data and all generated dose-response curves. Also plots a histogram of the EC50 estimates. The function makes use of the [plot.drFitSpline](#) function.

Usage

```
## S3 method for class 'drBootSpline'  
plot(x, pch = 1, colData = 1, colSpline = 1, cex = 1, ...)
```

Arguments

x	drFitSpline object
pch	Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from <code>plot.default</code> . The vector is recycled if the number of given data points differs from the length of the pch.
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of data points differ from the length of the colData.
colSpline	Vector, describing the color used for plotting the spline fit. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of spline fits differ from the length of the colSpline.
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from <code>plot.default</code> . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

`link{drBootSpline}`

Examples

```
# generate random data set
data <- ran.data(100,25)
time <- data$time
data <- data$data
# call grofit
TestRun<-grofit(time,data,TRUE,grofit.control(nboot.dr=50))
# plot dose-reponse curve sample
plot(TestRun$drFit$drBootSplines[[1]], colData=c(1,3),pch=1:4,cex=3:5,
      colSpline=c("tomato", "tomato1", "tomato2", "tomato3", "tomato4"))
```

plot.drFit

Generic plot function for drFit objects

Description

Generic plot function for drFit objects. Calls `drFitSpline` with respecting default values to visualize each dose response curve stored in the drFit object.

Usage

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


Arguments

x Object of class drFit.
 ... Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[drFit](#), [drFitSpline](#)

Examples

```
# generate random growth curve data set
foo <- ran.data(100, 25)
time <- foo$time
data <- foo$data
# fit dose response curves
drData <- gcFit(time,data)
# use the output of gcFit as an input for drFit
drFit.result <- drFit(summary(drData))
plot(drFit.result)
```

plot.drFitSpline

Generic plot function for drFitSpline objects

Description

Generic plot function for drFitSpline objects

Usage

```
## S3 method for class 'drFitSpline'
plot(x, add = FALSE, ec50line = TRUE,
      pch = 1, colSpline = 1, colData = 1,
      cex = 1, ...)
```

Arguments

x Object of class drFitSpline.
 add Logical, indicates whether the plot should be added to an existing plot device or plotted in a new window.
 ec50line Logical, indicates whether the EC50 value and respective response values should be visualized by a dotted line.
 pch Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from [plot.default](#). The vector is recycled if the number of given data points differs from the length of the pch.

colSpline	Vector, describing the color used for plotting the spline fit. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> .
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of data points differ from the length of the colData.
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from <code>plot.default</code> . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[drFitSpline](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(-0.5*(15-x)))+rnorm(30)/20
TestRun <- drFitSpline(x,y)
plot(TestRun, colData=c(1,3),pch=1:4,cex=3:5, colSpline=c("tomato"))
```

plot.gcBootSpline *Generic plot function for gcBootSpline objects*

Description

Generic plot function for gcBootSpline objects.

Usage

```
## S3 method for class 'gcBootSpline'
plot(x, pch = 1, colData = 1,
      colSpline = 1, cex = 1, ...)
```

Arguments

x	Object of class gcBootSpline.
pch	Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from <code>plot.default</code> . The vector is recycled if the number of given data points differs from the length of the pch.
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of data points differ from the length of the colData.

colSpline	Vector, describing the color used for plotting the spline fit. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of spline fits differ from the length of the colSpline.
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from <code>plot.default</code> . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[gcBootSpline](#)

Examples

```
x <- 1:30
y <- 1/(1+exp(0.5*(15-x)))+rnorm(30)/20
TestRun <- gcBootSpline(x,y,"ID",grofit.control(nboot.gc=50))
plot(TestRun, pch=2, cex=3, colData="steelblue",
      colSpline=c("orchid", "orchid1", "orchid2", "orchid3", "orchid4"))
```

plot.gcFit

Generic plot function for gcFit objects

Description

Generic plot function for gcFit objects.

Usage

```
## S3 method for class 'gcFit'
plot(x, opt = "m", raw = TRUE, slope = FALSE,
      pch = 1, colModel = 1, colSpline = 2,
      colData = 1, cex = 1, ...)
```

Arguments

x	Object of class gcFit.
opt	Character, indicating whether all available model fits ("s") or spline fits ("s") should be plotted.
raw	Logical, indicating whether the raw data should be plotted or not.
slope	Logical, indicating whether the tangent of maximal slope should be plotted or not.
pch	Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from <code>plot.default</code> . The vector is recycled if the number of given data points differs from the length of the pch.

colModel	Vector, describing the color used for plotting the spline fit. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> .
colSpline	Vector, describing the color used for plotting the spline fit. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> .
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of data points differ from the length of the colData.
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from <code>plot.default</code> . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[gcFit](#), [plot.gcFitSpline](#), [plot.gcFitModel](#)

Examples

```
data(grofit.time)
data(grofit.data)
TestRun <- gcFit(grofit.time, grofit.data)
dev.new(width=10,height=5)
par(mfrow=c(1,2))
plot(TestRun, opt="s", pch=1:7, cex=1, colSpline=2, colData=1:7)
title("Spline fit")
plot(TestRun, opt="m", pch=1:7, cex=1, colModel=1, colData=1:7)
title("Model fit")
par(mfrow=c(1,1))
```

plot.gcFitModel

Generic plot function for gcFitModel objects

Description

Generic plot function for gcFitModel objects.

Usage

```
## S3 method for class 'gcFitModel'
plot(x, add = FALSE, raw = TRUE,
      slope = TRUE, pch = 1,
      colData=1, colModel=1, cex = 1, ...)
```

Arguments

x	Object of class gcFitModel.
add	Logical, indicates whether the plot should be added to an existing plot device or plotted in a new window.
raw	Logical, indicating whether the raw data should be plotted or not.
slope	Logical, indicating whether the tangent of maximal slope should be plotted or not.
pch	Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from <code>plot.default</code> . The vector is recycled if the number of given data points differs from the length of the pch.
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> . The vector is recycled if the number of data points differ from the length of the colData.
colModel	Vector, describing the color used for plotting the spline fit and the tangent of maximal slope. Similar to the col option from <code>plot.default</code> . The values must be (numeric) from 0:8 or (character) an element of <code>colors()</code> .
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from <code>plot.default</code> . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[gcFitModel](#)

Examples

```
# generate random data set
data <- ran.data(100,25)
time <- data$time
data <- data$data
# run growth curve fit
TestRun <- gcFit(time,data)
plot(TestRun$gcFittedModels[[6]],
      colData=c("wheat", "wheat2", "wheat2", "wheat3", "wheat4"),
      colModel=1, pch=4:7, cex=2:5)
```

plot.gcFitSpline

Generic plot function for gcFitSpline objects

Description

Generic plot function for gcFitSpline objects.

Usage

```
## S3 method for class 'gcFitSpline'
plot(x, add = FALSE, raw = TRUE,
      slope = TRUE, pch = 1, colData=1,
      colSpline=2, cex = 1, ...)
```

Arguments

x	Object of class gcFitSpline.
add	Logical, indicates whether the plot should be added to an existing plot device or plotted in a new window.
raw	Logical, indicating whether the raw data should be plotted or not.
slope	Logical, indicating whether the tangent of maximal slope should be plotted or not.
pch	Numeric vector, describing plotting character used for plotting the raw data. Similar to the pch option from plot.default . The vector is recycled if the number of given data points differs from the length of the pch.
colData	Vector, describing the color used for plotting the raw data. Similar to the col option from plot.default . The values must be (numeric) from 0:8 or (character) an element of colors() . The vector is recycled if the number of data points differ from the length of the colData.
colSpline	Vector, describing the color used for plotting the spline fit and the tangent of maximal slope. Similar to the col option from plot.default . The values must be (numeric) from 0:8 or an element of colors() (character).
cex	Numeric vector, describing the character expansion used for plotting the raw data. Similar to the cex option from plot.default . The vector is recycled if the number of data points differ from the length of the cex.
...	Other graphical parameters may also be passed as arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

See Also

[gcFitSpline](#)

Examples

```
# generate random data set
data <- ran.data(100,25)
time <- data$time
data <- data$data
# run growth curve fit
TestRun <- gcFit(time,data)
plot(TestRun$gcFittedSpline[[3]],
      colData=c("wheat", "wheat2", "wheat2", "wheat3", "wheat4"),
      colSpline=1, pch=4:7, cex=2:5)
```

`ran.data`*Function to generate example data for grofit*

Description

The function calls the [gompertz](#) function to generate curves between zero and `t` and adds some random noise to the x- and y-axes. The three parameters given as input values will be slightly changed to produce different growth curves. The resulting datasets can be used to test the [grofit](#) function.

Usage

```
ran.data(d, t, mu = 1, lambda = 5, A = 15)
```

Arguments

<code>d</code>	Numeric value, number of data sets. If <code>d</code> is a vector, only the first entry is used.
<code>t</code>	Numeric value, number of time points per data set. If <code>t</code> is a vector, only the first entry is used.
<code>mu</code>	Numeric value, maximum slope. If <code>mu</code> is a vector, only the first entry is used.
<code>lambda</code>	Numeric value, lag-phase. If <code>lambda</code> is a vector, only the first entry is used.
<code>A</code>	Numeric value, maximum growth level. If <code>A</code> is a vector, only the first entry is used.

Details

The function generates two datasets according to the standard input of [grofit](#) and [gcFit](#).

Value

<code>time</code>	numeric matrix of size <code>dx</code> <code>t</code> , each row represent the time points for which growth data is simulated and stored in each row of data.
<code>data</code>	data.frame of size <code>dx</code> (<code>3+t</code>), 1. column, character as an experiment identifier; 2. column: character, additional information about respecting experiment; 3. column: concentration of substrate of a compound under which the experiment is obtained; 4.-(<code>3+t</code>). column: growth data corresponding to the time points in time.

See Also

[grofit.time](#), [grofit.data](#)

`richards`*Richards growth law*

Description

The function calculates the values of the Richards growth curve for given time points.

Usage

```
richards(time, A, mu, lambda, addpar)
```

Arguments

<code>time</code>	Time points (x-axes) for which the function values will be returned.
<code>A</code>	Maximum of the curve. If a vector is provided only the first entry is used.
<code>mu</code>	Maximum slope. If a vector is provided only the first entry is used.
<code>lambda</code>	Lag-phase. If a vector is provided only the first entry is used.
<code>addpar</code>	Numeric vector of size one, corresponding to shape parameter ν .

Details

The function follows the parametrization

$$y(t) = A \cdot \left[1 + \nu \exp \left(1 + \nu + \frac{\mu}{A} (1 + \nu)^{1+1/\nu} (\lambda - t) \right) \right]^{-1/\nu}$$

Value

Numeric vector giving the values of Richards growth law for numeric input vector `time`.

Examples

```
x <- 1:30
y <- richards(x, 10, 2, 5, 3)
plot(x,y)
```

summary.drBootSpline *Generic summary function for drBootSpline object*

Description

Generic summary function for drBootSpline object. Generates a data.frame of mean EC50, standard deviation and confidence intervals from a bootstrap sample of dose-response curves.

Usage

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

Arguments

object	Object of class drBootSpline.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame	1.-2. column: numeric, mean and standard dev. from bootstrap sample; 3.-6. column: 90 and 95 percent confidence interval from bootstrap sample; 7. column: mean EC50 of bootstrap sample in original scale; 8.-11. 90 and 95 percent confidence interval in original scale.
------------	---

See Also

[drBootSpline](#), [drFit](#), [summary.drFit](#)

summary.drFit *Generic summary function for drFit object*

Description

Generic summary function for drFit object. Extracts the data.frame drTable generated by [drFit](#).

Usage

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

Arguments

object Object of class drFit.

... Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame 1. column: character, naming the experiment; 2. column: logical, indicating wether a logarithmic transformation was applied to concentration data; 3. column, logical, indicating wether a logarithmic transformation was applied to growth parameter; 4. column, numeric, number of bootstrap samples used; 5.-8. column: numeric, estimated EC50 value and respecting reponse value (5.-6.), also in original scale if a transformation is applied (7.-8.); 9.-10. column: numeric, mean and standard dev. from bootstrap sample; 11.-14. column: 90 and 95 percent confidence interval from bootstrap sample; 15. column: mean EC50 of bootstrap sample in original scale; 16.-19. 90 and 95 percent confidence interval in original scale

See Also

[drFit](#), [summary.drFitSpline](#), [summary.drBootSpline](#)

summary.drFitSpline *Generic summary function for drFitSpline object*

Description

Generic summary function for drFitSpline object. Generates a data.frame with EC50 value and respecting reponse value in transformed and original scale.

Usage

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

Arguments

object Object of class drFitSpline.

... Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame 1.-2. column: numeric, estimated EC50 value and respecting reponse value; 3.-4. column: EC50 and response in original scale when a transformation is applied.

See Also

[drFit](#), [summary.drFitSpline](#), [summary.drBootSpline](#)

summary.gcBootSpline *Generic summary function for gcBootSpline object*

Description

Generic summary function for gcBootSpline object. Generates a data.frame including the mean, standard deviation and confidence intervals for the characteristic growth values from a bootstrap sample of a growth curve.

Usage

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

Arguments

object	Object of class drBootSpline.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame	1.-4. column: mean of μ , λ , A and <i>integral</i> from bootstrap sample; 5.-8. column: respecting standard deviation; 9.-16. column: 90 percent confidence intervals for μ , λ , A and <i>integral</i> ; 17.-24. column: 95 percent confidence intervals for μ , λ , A and <i>integral</i> .
------------	--

See Also

[gcBootSpline](#), [gcFit](#)

summary.gcFit *Generic summary function for gcFit object*

Description

Generic summary function for gcFit object.

Usage

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

Arguments

object Object of class `gcFit`.

... Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Details

Columns 1.-8. are generated by `gcFit`; columns 9.-27. are generated by `summary.gcFitModel`; columns 28.-31. are generated by `summary.gcFitSpline`; columns 32.-55. are generated by `summary.gcBootSpline`.

Value

data.frame 1. column: character, naming the experiment for which a dose-response curve shall be estimated; 2. column: character, additional information about the experiment; 3. column: numeric, concentration of substrate; 4. column: logical, reliability flag, 5. column: character, naming a parametric model, 6. column: logical, indicating wether a logarithmic transformation was applied to the time points of the growth data; 7. column: logical, indicating wether a logarithmic transformation was applied to the growth values of the growth data; 8. column: number of bootstrap samples used; 9.-12. column: estimates of μ , λ , A and *integral* from growth curve fit; 13.-15. column: respecting standard deviation for μ , λ , A ; 16.-21. column: 90 percent confidence intervals for μ , λ , A ; 22.-27. column: 95 percent confidence intervals for μ , λ , A ; 28.-31. column: estimates of μ , λ , A and *integral* from growth curve spline fit; 32.-35. column: mean of μ , λ , A and *integral* from bootstrap sample; 36.-39. column: respecting standard deviation; 40.-47. column: 90 percent confidence intervals for μ , λ , A and *integral*; 48.-55. column: 95 percent confidence intervals for μ , λ , A and *integral*.

See Also

[gcBootSpline](#), [gcFit](#), [gcFitSpline](#), [gcFitModel](#)

summary.gcFitModel *Generic summary function for gcFitModel object*

Description

Generic summary function for `gcFitModel` object.

Usage

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

Arguments

object Object of class gcFitModel.
 ... Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame 1.-4. column: estimates of μ , λ , A and *integral* from growth curve fit; 5.-7. column: respecting standard deviation for μ , λ , A ; 8.-13. column: 90 percent confidence intervals for μ , λ , A ; 17.-24. column: 95 percent confidence intervals for μ , λ , A .

See Also

[gcFitModel](#), [gcFit](#)

summary.gcFitSpline *Generic summary function for gcFitSpline object*

Description

Generic summary function for gcFitSpline object.

Usage

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

Arguments

object Object of class gcFitSpline.
 ... Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

Value

data.frame 1.-4. column: estimates of μ , λ , A and *integral* from growth curve fit.

See Also

[gcFitSpline](#), [gcFit](#)

Index

*Topic **datasets**

grofit.data, 18

grofit.time, 18

*Topic **package**

grofit-package, 2

drBootSpline, 3, 5, 6, 33

drFit, 5, 7, 10, 15, 17, 25, 33–35

drFitSpline, 3–6, 6, 24–26

gcBootSpline, 8, 9, 13, 27, 35, 36

gcFit, 5, 9, 9, 15, 17, 28, 31, 35–37

gcFitModel, 9, 10, 11, 13, 19–21, 29, 36, 37

gcFitSpline, 8–12, 12, 30, 36, 37

gompertz, 12, 13, 19, 31

gompertz.exp, 12, 14, 20

grofit, 10, 15, 31

grofit-package, 2

grofit.control, 4, 5, 7–9, 11, 12, 15, 16

grofit.data, 10, 16, 18, 18, 31

grofit.time, 10, 16, 18, 18, 31

initgompertz, 11, 19

initgompertz.exp, 19

initlogistic, 11, 20

initrichards, 21

logistic, 12, 21, 22

low.integrate, 23

lowess, 11, 13

nls, 11, 12, 19–21

plot.default, 24–30

plot.drBootSpline, 4, 23

plot.drFit, 6, 24

plot.drFitSpline, 7, 23, 25

plot.gcBootSpline, 9, 26

plot.gcFit, 10, 27

plot.gcFitModel, 12, 28, 28

plot.gcFitSpline, 13, 28, 29

ran.data, 10, 31

richards, 12, 21, 32

smooth.spline, 7, 13, 17, 23

summary.drBootSpline, 4, 33, 34, 35

summary.drFit, 6, 33, 33

summary.drFitSpline, 7, 34, 34, 35

summary.gcBootSpline, 9, 35, 36

summary.gcFit, 10, 17, 35

summary.gcFitModel, 12, 36, 36

summary.gcFitSpline, 13, 36, 37