

# Package ‘QurvE’

October 28, 2022

**Title** Robust and user-friendly analysis of growth curves

**Version** 0.9.2.3

**Description** With QurvE, growth curves and fluorescence data can be analyzed in high throughput with three methods: linear regression, fitting of growth models, smooth spline fit. User-friendly wrapper functions allow complete data analysis workflows to be executed in a single step. The results of these workflows are summarized in detailed reports as well as in R data containers that can be intuitively navigated. A shiny application provides access to all features of the package without requiring programming knowledge.

**License** GPL (>= 3)

**URL** <https://github.com/NicWir/QurvE>

**BugReports** <https://github.com/NicWir/QurvE/issues>

**Depends** dplyr,  
methods,  
R (>= 4.0),  
stringr,  
tidyr

**Imports** doParallel,  
drc,  
DT,  
foreach,  
ggnewscale,  
ggplot2,  
ggpubr,  
kableExtra,  
knitr,  
labeling,  
magrittr,  
minpack.lm,  
plyr,  
RColorBrewer,  
readxl,  
rmarkdown,  
scales,  
shiny,  
stats,  
utils

**Suggests** bookdown,

Cairo,  
htmltools,  
prettydoc,  
rlang,  
shinyBS,  
shinycssloaders,  
shinyFiles,  
shinyjs,  
shinysurveys,  
shinythemes,  
tibble,  
tinytex

**VignetteBuilder** knitr

**biocViews**

**Encoding** UTF-8

**GithubRepo** QurvE

**GithubUsername** NicWir

**NeedsCompilation** yes

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.1

**Collate** 'general\_misc\_utils.R'

'fluorescence\_computation.R'

'fluorescence\_plots.R'

'fluorescence\_summaries.R'

'growth\_computation.R'

'growth\_plots.R'

'growth\_summaries.R'

'shiny\_app\_functions.R'

## R topics documented:

.onAttach . . . . .	4
colFmt . . . . .	4
fl.control . . . . .	4
fl.drFit . . . . .	7
fl.drFitModel . . . . .	8
fl.report . . . . .	9
fl.workflow . . . . .	10
flBootSpline . . . . .	14
flFit . . . . .	15
flFitLinear . . . . .	17
flFitSpline . . . . .	19
grofit.param . . . . .	21
growth.control . . . . .	21
growth.drBootSpline . . . . .	24
growth.drFit . . . . .	25
growth.drFitModel . . . . .	26
growth.drFitSpline . . . . .	27

growth.gcBootSpline . . . . .	28
growth.gcFit . . . . .	29
growth.gcFitLinear . . . . .	30
growth.gcFitModel . . . . .	34
growth.gcFitSpline . . . . .	35
growth.report . . . . .	37
growth.workflow . . . . .	38
match_arg . . . . .	42
numberInput . . . . .	42
parse_data . . . . .	44
plot.drBootSpline . . . . .	45
plot.drFit . . . . .	46
plot.drFitFLModel . . . . .	48
plot.drFitModel . . . . .	49
plot.drFitSpline . . . . .	51
plot.dr_parameter . . . . .	52
plot.dual . . . . .	54
plot.flBootSpline . . . . .	56
plot.flFitLinear . . . . .	57
plot.flFitRes . . . . .	59
plot.flFitSpline . . . . .	62
plot.gcBootSpline . . . . .	63
plot.gcFitLinear . . . . .	65
plot.gcFitModel . . . . .	66
plot.gcFitSpline . . . . .	67
plot.grodata . . . . .	69
plot.grofit . . . . .	71
plot.parameter . . . . .	73
read_data . . . . .	75
read_file . . . . .	77
run_app . . . . .	77
summary.drBootSpline . . . . .	78
summary.drFit . . . . .	78
summary.drFitFLModel . . . . .	79
summary.drFitModel . . . . .	79
summary.drFitSpline . . . . .	80
summary.flBootSpline . . . . .	80
summary.flFit . . . . .	81
summary.flFitLinear . . . . .	81
summary.flFitSpline . . . . .	82
summary.gcBootSpline . . . . .	82
summary.gcFit . . . . .	83
summary.gcFitLinear . . . . .	83
summary.gcFitModel . . . . .	84
summary.gcFitSpline . . . . .	84
table_group_fluorescence_linear . . . . .	85
table_group_fluorescence_spline . . . . .	85
table_group_growth_linear . . . . .	86
table_group_growth_model . . . . .	86
table_group_growth_spline . . . . .	87
xgx_breaks_log10 . . . . .	87
xgx_minor_breaks_log10 . . . . .	88

**Index****89**

---

<code>.onAttach</code>	<i>Run upon attaching package QurvE</i>
------------------------	---

---

**Description**

Changes debug option for package rgl to avoid Rstudio crashing upon attaching it and prints welcome message

**Usage**

```
.onAttach(libname, pkgname)
```

**Arguments**

<code>libname</code>	library name
<code>pkgname</code>	package name

---

<code>colFmt</code>	<i>Format font color for Markdown reports</i>
---------------------	---

---

**Description**

`colFmt` formats the input depending on PDF or HTML output to give colored text in reports.

**Usage**

```
colFmt(x, color)
```

**Arguments**

<code>x</code>	A character string. The text to be colored.
<code>color</code>	(Character) A color.

---

fl.control	Create a fl.control object.
------------	-----------------------------

---

## Description

A fl.control object is required to perform various computations on fluorescence data stored within grodata objects (created with [read\\_data](#) or [parse\\_data](#)). A fl.control object is created automatically as part of [fl.workflow](#).

## Usage

```
fl.control(
  fit.opt = c("l", "s"),
  x_type = c("density", "time"),
  norm_fl = TRUE,
  t0 = 0,
  min.density = NA,
  log.x.lin = FALSE,
  log.x.spline = FALSE,
  log.y.lin = FALSE,
  log.y.spline = FALSE,
  lin.h = NULL,
  lin.R2 = 0.97,
  lin.RSD = 0.05,
  lin.dY = 0.05,
  dr.parameter = "max_slope.spline",
  dr.method = c("model", "spline"),
  dr.have.atleast = 5,
  smooth.dr = NULL,
  log.x.dr = FALSE,
  log.y.dr = FALSE,
  nboot.dr = 0,
  biphasic = FALSE,
  interactive = FALSE,
  nboot.fl = 0,
  smooth.fl = 0.75,
  growth.thresh = 1.5,
  suppress.messages = FALSE,
  neg.nan.act = FALSE,
  clean.bootstrap = TRUE
)
```

## Arguments

fit.opt	(Character or vector of strings) Indicates whether the program should perform a linear regression ("l") and/or spline fit ("s"). Default: fit.opt = c("l", "s").
x_type	(Character) Which data type shall be used as independent variable? Options are 'density' and 'time'.
norm_fl	(Logical) use normalized (to density) fluorescence data in fits. Has an effect only when x_type = 'time'

<code>t0</code>	(Numeric) Minimum time value considered for linear and spline fits.
<code>min.density</code>	(Numeric) Indicate whether only values above a certain threshold should be considered for linear regressions or spline fits.
<code>log.x.lin</code>	(Logical) Indicates whether $\ln(x+1)$ should be applied to the independent variable for <i>linear</i> fits. Default: FALSE.
<code>log.x.spline</code>	(Logical) Indicates whether $\ln(x+1)$ should be applied to the independent variable for <i>spline</i> fits. Default: FALSE.
<code>log.y.lin</code>	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the fluorescence data for <i>linear</i> fits. Default: FALSE
<code>log.y.spline</code>	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the fluorescence data for <i>spline</i> fits. Default: FALSE
<code>lin.h</code>	(Numeric) Manually define the size of the sliding window used in <a href="#">flFitLinear</a> . If NULL, h is calculated for each samples based on the number of measurements in the fluorescence increase phase of the plot.
<code>lin.R2</code>	(Numeric) $R^2$ threshold for <a href="#">flFitLinear</a> .
<code>lin.RSD</code>	(Numeric) Relative standard deviation (RSD) threshold for the calculated slope in <a href="#">flFitLinear</a> .
<code>lin.dY</code>	(Numeric) Threshold for the minimum fraction of density increase a linear regression window should cover. Default: 0.05 (5%).
<code>dr.parameter</code>	(Character or numeric) The response parameter in the output table to be used for creating a dose response curve. See <a href="#">fl.drFit</a> for further details. Default: "max_slope.spline", which represents the maximum slope of the spline fit. Typical options include: "max_slope.linfit", "dY.linfit", "max_slope.spline", and "dY.spline".
<code>dr.method</code>	(Character) Perform either a smooth spline fit on response parameter vs. concentration data ("spline") or fit a biosensor response model (proposed by Meyer et al., 2019).
<code>dr.have.atleast</code>	(Numeric) 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>smooth.dr</code>	(Numeric) Smoothing parameter used in the spline fit by <code>smooth.spline</code> during dose response curve estimation. Usually (not necessary) in (0; 1]. See <a href="#">smooth.spline</a> for further details. Default: NULL.
<code>log.x.dr</code>	(Logical) Indicates whether $\ln(x+1)$ should be applied to the concentration data of the dose response curves. Default: FALSE.
<code>log.y.dr</code>	(Logical) Indicates whether $\ln(y+1)$ should be applied to the response data of the dose response curves. Default: FALSE.
<code>nboot.dr</code>	(Numeric) Defines the number of bootstrap samples for EC50 estimation. Use <code>nboot.dr = 0</code> to disable bootstrapping. Default: 0.
<code>biphasic</code>	(Logical) Shall <a href="#">flFitLinear</a> and <a href="#">flFitSpline</a> try to extract fluorescence parameters for two different phases (as observed with, e.g., regulator-promoter systems with varying response in different growth stages) (TRUE) or not (FALSE)?
<code>interactive</code>	(Logical) Controls whether the fit for each sample and method is controlled manually by the user. If TRUE, each fit is visualized in the <i>Plots</i> pane and the user can adjust fitting parameters and confirm the reliability of each fit per sample. Default: TRUE.

nboot.fl	(Numeric) Number of bootstrap samples used for nonparametric curve fitting with <code>flBootSpline</code> . Use <code>nboot.fl = 0</code> to disable the bootstrap. Default: 0
smooth.fl	(Numeric) Parameter describing the smoothness of the spline fit; usually (not necessary) within (0;1]. <code>smooth.gc=NULL</code> causes the program to query an optimal value via cross validation techniques. Especially for datasets with few data points the option NULL might cause a too small smoothing parameter. This can result a too tight fit that is susceptible to measurement errors (thus overestimating slopes) or produce an error in <code>smooth.spline</code> or lead to overfitting. The usage of a fixed value is recommended for reproducible results across samples. See <code>smooth.spline</code> for further details. Default: 0.55
growth.thresh	(Numeric) Define a threshold for growth. Only if any density value in a sample is greater than <code>growth.thresh</code> (default: 1.5) times the start density, further computations are performed. Else, a message is returned.
suppress.messages	(Logical) Indicates whether messages (information about current fluorescence curve, EC50 values etc.) should be displayed (FALSE) or not (TRUE). This option is meant to speed up the high-throughput processing data. Note: warnings are still displayed. Default: FALSE.
neg.nan.act	(Logical) Indicates whether the program should stop when negative fluorescence values or NA values appear (TRUE). Otherwise, the program removes these 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.

## Value

Generates a list with all arguments described above as entries.

## References

Meyer, A.J., Segall-Shapiro, T.H., Glassey, E. et al. *Escherichia coli* “Marionette” strains with 12 highly optimized small-molecule sensors. Nat Chem Biol 15, 196–204 (2019). DOI: 10.1038/s41589-018-0168-3

---

fl.drFit	<i>Fit a biosensor model (Meyer et al., 2019) to response vs. concentration data</i>
----------	--

---

## Description

Fit a biosensor model (Meyer et al., 2019) to response vs. concentration data

## Usage

```
fl.drFit(FitData, control = fl.control())
```

**Arguments**

FitData	A dataframe containing the data for the dose-response model estimation. Such table of class <code>flTable</code> can be obtained by running <code>flFit</code> with <code>dr.method = "model"</code> as argument in the <code>fl.control</code> object.
control	A <code>fl.control</code> object created with <code>fl.control</code> , defining relevant fitting options.

**Value**

An object of class `drFit`.

<code>raw.data</code>	Data that passed to the function as <code>flTable</code> .
<code>drTable</code>	Dataframe containing condition identifiers, fit options, and results of the dose-response analysis.
<code>drFittedModels</code>	List of all <code>drFitModel</code> objects generated by the call of <code>fl.drFitModel</code> for each distinct experiment.
<code>control</code>	Object of class <code>fl.control</code> created with the call of <code>fl.control</code> .

**References**

Meyer, A.J., Segall-Shapiro, T.H., Glassey, E. et al. *Escherichia coli* “Marionette” strains with 12 highly optimized small-molecule sensors. *Nat Chem Biol* 15, 196–204 (2019). DOI: 10.1038/s41589-018-0168-3

---

<code>fl.drFitModel</code>	<i>Perform a biosensor model fit on response vs. concentration data of a single sample.</i>
----------------------------	---

---

**Description**

`fl.drFitModel` fits the biosensor model proposed by Meyer et al. (2019) to the provided response (e.g., `max_slope.spline` vs. concentration data to determine the leakiness, sensitivity, induction fold-change, and cooperativity.

**Usage**

```
fl.drFitModel(conc, test, drID = "undefined", control = fl.control())
```

**Arguments**

<code>conc</code>	Vector of concentration values.
<code>test</code>	Vector of response parameter values of the same length as <code>conc</code> .
<code>drID</code>	(Character) The name of the analyzed condition
<code>control</code>	A <code>fl.control</code> object created with <code>fl.control</code> , defining relevant fitting options.



**Value**

A `drFitFLModel` object.

<code>raw.conc</code>	Raw data provided to the function as <code>conc</code> .
<code>raw.test</code>	Raw data for the response parameter provided to the function as <code>test</code> .
<code>drID</code>	(Character) Identifies the tested condition
<code>fit.conc</code>	Fitted concentration values.
<code>fit.test</code>	Fitted response values.
<code>model</code>	<code>nls</code> object generated by the <code>nlsLM</code> function.
<code>parameters</code>	List of parameters estimated from dose response curve fit. <ul style="list-style-type: none"> <li>• <code>yEC50</code>: Response value related to EC50.</li> <li>• <code>y.min</code>: Minimum fluorescence ('leakiness', if lowest concentration is 0).</li> <li>• <code>y.max</code>: Maximum fluorescence.</li> <li>• <code>fc</code>: Fold change (<code>y.max</code> divided by <code>y.min</code>).</li> <li>• <code>K</code>: Concentration at half-maximal response ('sensitivity').</li> <li>• <code>n</code>: Cooperativity.</li> <li>• <code>yEC50.orig</code>: Response value for EC50 in original scale, if a transformation was applied.</li> <li>• <code>K.orig</code>: K in original scale, if a transformation was applied.</li> <li>• <code>test.nm</code>: Test identifier extracted from <code>test</code>.</li> </ul>
<code>fitFlag</code>	(Logical) Indicates whether a spline could fitted successfully to data.
<code>reliable</code>	(Logical) Indicates whether the performed fit is reliable (to be set manually).
<code>control</code>	Object of class <code>fl.control</code> created with the call of <code>fl.control</code> .

Use `plot.drFitModel` to visualize the model fit.

**References**

Meyer, A.J., Segall-Shapiro, T.H., Glassey, E. et al. *Escherichia coli* "Marionette" strains with 12 highly optimized small-molecule sensors. Nat Chem Biol 15, 196–204 (2019). DOI: 10.1038/s41589-018-0168-3

---

<code>fl.report</code>	Create a PDF and HTML report with results from a fluorescence analysis workflow
------------------------	---

---

**Description**

`fl.report` requires a `flFitRes` object and creates a report in PDF and HTML format that summarizes all results obtained.

## Usage

```
fl.report(
  flFitRes,
  out.dir = NULL,
  out.nm = NULL,
  ec50 = FALSE,
  format = c("pdf", "html"),
  export = FALSE,
  ...
)
```

## Arguments

flFitRes	A grofit object created with <a href="#">fl.workflow</a> .
out.dir	(Character) The path or name of the folder in which the report files are created. If NULL, the folder will be named with a combination of "Report.fluorescence_" and the current date and time.
out.nm	Character or NULL Define the name of the report files. If NULL, the files will be named with a combination of "FluorescenceReport_" and the current date and time.
ec50	(Logical) Display results of dose-response analysis (TRUE) or not (FALSE).
format	(Character) Define the file format for the report, PDF ('pdf') and/or HTML ('html'). Default: (c('pdf', 'html'))
export	(Logical) Shall all plots generated in the report be exported as individual PDF and PNG files TRUE or not FALSE?
...	Further arguments passed to create a report. Currently required: <ul style="list-style-type: none"> <li>• mean.grp: Define groups to combine into common plots in the report based on sample identifiers. Partial matches with sample/group names are accepted. Can be "all", a vector of strings, or a list of string vectors. Note: The maximum number of sample groups (with unique condition/concentration indicators) is 50. If you have more than 50 groups, option "all" will produce the error ! Insufficient values in manual scale. [Number] needed but only 50 provided.</li> <li>• mean.conc: Define concentrations to combine into common plots in the report. Can be a numeric vector, or a list of numeric vectors.</li> </ul>

---

fl.workflow	<i>Run a complete fluorescence curve analysis and dose-reponse analysis workflow.</i>
-------------	---

---

## Description

fl.workflow runs [fl.control](#) to create a fl.control object and then performs all computational fitting operations based on the user input. Finally, if desired, a final report is created in PDF or HTML format that summarizes all results obtained.

**Usage**

```

fl.workflow(
  grodata = NULL,
  time = NULL,
  density = NULL,
  fl_data = NULL,
  ec50 = FALSE,
  mean.grp = NA,
  mean.conc = NA,
  fit.opt = c("l", "s"),
  x_type = c("density", "time"),
  norm_fl = TRUE,
  t0 = 0,
  min.density = 0,
  log.x.lin = FALSE,
  log.x.spline = FALSE,
  log.y.lin = FALSE,
  log.y.spline = FALSE,
  lin.h = NULL,
  lin.R2 = 0.97,
  lin.RSD = 0.05,
  lin.dY = 0.05,
  biphasic = FALSE,
  interactive = FALSE,
  dr.parameter = "max_slope.spline",
  dr.method = c("model", "spline"),
  dr.have.atleast = 5,
  smooth.dr = NULL,
  log.x.dr = FALSE,
  log.y.dr = FALSE,
  nboot.dr = 0,
  nboot.fl = 0,
  smooth.fl = 0.75,
  growth.thresh = 1.5,
  suppress.messages = FALSE,
  neg.nan.act = FALSE,
  clean.bootstrap = TRUE,
  report = c("pdf", "html"),
  out.dir = NULL,
  out.nm = NULL,
  export.fig = FALSE,
  export.res = FALSE,
  ...
)

```

**Arguments**

grodata	A grodata object created with <a href="#">read_data</a> or <a href="#">parse_data</a> , containing fluorescence data and data for the independent variable (i.e., time or density).
time	(optional) A matrix containing time values for each sample (if a fl_data dataframe is provided as separate argument).

density	(optional) A dataframe containing growth data (if a fl_data matrix is provided as separate argument).
fl_data	(optional) A dataframe containing fluorescence data (if a time matrix or density dataframe is provided as separate argument).
ec50	(Logical) Perform dose-response analysis (TRUE) or not (FALSE).
mean.grp	("all", a string vector, or a list of string vectors) Define groups to combine into common plots in the final report based on sample identifiers (if report == TRUE). Partial matches with sample/group names are accepted. Note: The maximum number of sample groups (with unique condition/concentration indicators) is 50. If you have more than 50 groups, option "all" will produce the error ! Insufficient values in manual scale. [Number] needed but only 50 provided.
mean.conc	(A numeric vector, or a list of numeric vectors) Define concentrations to combine into common plots in the final report (if report == TRUE).
fit.opt	(Character or character vector) Indicates whether the program should perform a linear regression ("l"), model fit ("m"), spline fit ("s"), or all ("a"). Combinations can be freely chosen by providing a character vector, e.g. fit.opt = c("l", "s") Default: fit.opt = c("l", "s").
x_type	(Character) Which data type shall be used as independent variable? Options are 'density' and 'time'.
norm.fl	(Logical) use normalized (to density) fluorescence data in fits. Has an effect only when x_type = 'time'
t0	(Numeric) Minimum time value considered for linear and spline fits.
min.density	(Numeric) Indicate whether only values above a certain threshold should be considered for linear regressions or spline fits.
log.x.lin	(Logical) Indicates whether $\ln(x+1)$ should be applied to the independent variable for <i>linear</i> fits. Default: FALSE.
log.x.spline	(Logical) Indicates whether $\ln(x+1)$ should be applied to the independent variable for <i>spline</i> fits. Default: FALSE.
log.y.lin	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the fluorescence data for <i>linear</i> fits. Default: FALSE
log.y.spline	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the fluorescence data for <i>spline</i> fits. Default: FALSE
lin.h	(Numeric) Manually define the size of the sliding window used in <a href="#">flFitLinear</a> . If NULL, h is calculated for each samples based on the number of measurements in the fluorescence increase phase of the plot.
lin.R2	(Numeric) $R^2$ threshold for <a href="#">flFitLinear</a> .
lin.RSD	(Numeric) Relative standard deviation (RSD) threshold for the calculated slope in <a href="#">flFitLinear</a> .
lin.dY	(Numeric) Threshold for the minimum fraction of density increase a linear regression window should cover. Default: 0.05 (5%).
biphasic	(Logical) Shall <a href="#">flFitLinear</a> and <a href="#">flFitSpline</a> try to extract fluorescence parameters for two different phases (as observed with, e.g., regulator-promoter systems with varying response in different growth stages) (TRUE) or not (FALSE)?
interactive	(Logical) Controls whether the fit for each sample and method is controlled manually by the user. If TRUE, each fit is visualized in the <i>Plots</i> pane and the user can adjust fitting parameters and confirm the reliability of each fit per sample. Default: TRUE.

<code>dr.parameter</code>	(Character or numeric) The response parameter in the output table to be used for creating a dose response curve. See <a href="#">fl.drFit</a> for further details. Default: <code>"max_slope.spline"</code> , which represents the maximum slope of the spline fit. Typical options include: <code>"max_slope.linfit"</code> , <code>"dY.linfit"</code> , <code>"max_slope.spline"</code> , and <code>"dY.spline"</code> .
<code>dr.method</code>	(Character) Perform either a smooth spline fit on response parameter vs. concentration data ( <code>"spline"</code> ) or fit a biosensor response model (proposed by Meyer et al., 2019).
<code>dr.have.atleast</code>	(Numeric) 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>smooth.dr</code>	(Numeric) Smoothing parameter used in the spline fit by <code>smooth.spline</code> during dose response curve estimation. Usually (not necessary) in (0; 1]. See <a href="#">smooth.spline</a> for further details. Default: NULL.
<code>log.x.dr</code>	(Logical) Indicates whether $\ln(x+1)$ should be applied to the concentration data of the dose response curves. Default: FALSE.
<code>log.y.dr</code>	(Logical) Indicates whether $\ln(y+1)$ should be applied to the response data of the dose response curves. Default: FALSE.
<code>nboot.dr</code>	(Numeric) Defines the number of bootstrap samples for EC50 estimation. Use <code>nboot.dr = 0</code> to disable bootstrapping. Default: 0.
<code>nboot.fl</code>	(Numeric) Number of bootstrap samples used for nonparametric curve fitting with <a href="#">flBootSpline</a> . Use <code>nboot.fl = 0</code> to disable the bootstrap. Default: 0
<code>smooth.fl</code>	(Numeric) Parameter describing the smoothness of the spline fit; usually (not necessary) within (0;1]. <code>smooth.gc=NULL</code> causes the program to query an optimal value via cross validation techniques. Especially for datasets with few data points the option NULL might cause a too small smoothing parameter. This can result a too tight fit that is susceptible to measurement errors (thus overestimating slopes) or produce an error in <a href="#">smooth.spline</a> or lead to overfitting. The usage of a fixed value is recommended for reproducible results across samples. See <a href="#">smooth.spline</a> for further details. Default: 0.55
<code>growth.thresh</code>	(Numeric) Define a threshold for growth. Only if any density value in a sample is greater than <code>growth.thresh</code> (default: 1.5) times the start density, further computations are performed. Else, a message is returned.
<code>suppress.messages</code>	(Logical) Indicates whether messages (information about current fluorescence curve, EC50 values etc.) should be displayed (FALSE) or not (TRUE). This option is meant to speed up the high-throughput processing data. Note: warnings are still displayed. Default: FALSE.
<code>neg.nan.act</code>	(Logical) Indicates whether the program should stop when negative fluorescence values or NA values appear (TRUE). Otherwise, the program removes these values silently (FALSE). Improper values may be caused by incorrect data or input errors. Default: FALSE.
<code>clean.bootstrap</code>	(Logical) Determines if negative values which occur during bootstrap should be removed (TRUE) or kept (FALSE). Note: Infinite values are always removed. Default: TRUE.
<code>report</code>	(Character or NULL) Create a PDF ( <code>'pdf'</code> ) and/or HTML ( <code>'html'</code> ) report after running all computations. Define NULL if no report should be created. Default: <code>(c('pdf', 'html'))</code>

out.dir	Character or NULL Define the name of a folder in which all result files are stored. If NULL, the folder will be named with a combination of "Report.fluorescence_" and the current date and time.
out.nm	Character or NULL Define the name of the report files. If NULL, the files will be named with a combination of "Fluorescenceport_" and the current date and time.
export.fig	(Logical) Export all figures created in the report as separate PNG and PDF files (TRUE) or not (FALSE).
export.res	(Logical) Create tab-separated TXT files containing calculated parameters and dose-response analysis results as well as an .RData file for the resulting flFitRes object at the end of the workflow.
...	Further arguments passed to the shiny app.

### Value

A flFitRes object that contains all computation results, compatible with various plotting functions of the QurvE package and with [fl.report](#).

time	Raw time matrix passed to the function as time (if no grofit object is provided. Else, extracted from grofit).
data	Raw data dataframe passed to the function as grodata.
flFit	flFit object created with the call of <a href="#">flFit</a> on fluorescence data.
drFit	drFit or drFitfl object created with the call of <a href="#">growth.drFit</a> or <a href="#">fl.drFit</a> for fluorescence data (based on the dr.method argument in control; see <a href="#">fl.control</a> ).
expdesign	Experimental design table inherited from grodata or created from the identifier columns (columns 1-3) in data.
control	Object of class fl.control created with the call of <a href="#">fl.control</a> .

---

flBootSpline

*flBootSpline: Function to generate a bootstrap*


---

### Description

fl.gcBootSpline resamples the fluorescence-'x' value pairs in a dataset with replacement and performs a spline fit for each bootstrap sample.

### Usage

```
flBootSpline(
  time = NULL,
  density = NULL,
  fl_data,
  ID = "undefined",
  control = fl.control()
)
```

**Arguments**

time	Vector of the independent variable: time (if <code>x_type = 'time'</code> in <code>fl.control</code> object).
density	Vector of the independent variable: density (if <code>x_type = 'density'</code> in <code>fl.control</code> object).
fl_data	Vector of dependent variable: fluorescence.
ID	(Character) The name of the analyzed sample.
control	A <code>fl.control</code> object created with <a href="#">fl.control</a> , defining relevant fitting options.

**Value**

A `gcBootSpline` object containing a distribution of fluorescence parameters and a `flFitSpline` object for each bootstrap sample. Use [plot.gcBootSpline](#) to visualize all bootstrapping splines as well as the distribution of physiological parameters.

raw.x	Raw time values provided to the function as time.
raw.fl	Raw density data provided to the function as data.
ID	(Character) Identifies the tested sample.
boot.x	Table of time values per column, resulting from each spline fit of the bootstrap.
boot.fl	Table of density values per column, resulting from each spline fit of the bootstrap.
boot.flSpline	List of <code>flFitSpline</code> object, created by <a href="#">flFitSpline</a> for each resample of the bootstrap.
lambda	Vector of estimated lambda (lag time) values from each bootstrap entry.
max_slope	Vector of estimated max_slope (maximum slope) values from each bootstrap entry.
A	Vector of estimated A (maximum fluorescence) values from each bootstrap entry.
integral	Vector of estimated integral values from each bootstrap entry.
bootFlag	(Logical) Indicates the success of the bootstrapping operation.
control	Object of class <code>fl.control</code> containing list of options passed to the function as control.

**See Also**

Other fluorescence fitting functions: [flFitSpline\(\)](#), [flFit\(\)](#)

---

flFit	<i>Perform a fluorescence curve analysis on all samples in the provided dataset.</i>
-------	--

---

### Description

flFit performs all computational fluorescence fitting operations based on the user input.

### Usage

```
flFit(fl_data, time = NULL, density = NULL, control = fl.control(), ...)
```

### Arguments

fl_data	Either... <ol style="list-style-type: none"> <li>1. a grodata object created with <a href="#">read_data</a> or <a href="#">parse_data</a>,</li> <li>2. a list containing a 'time' matrix (for x_type == "time") or 'density' dataframe (for x_type == "density") and a 'fluorescence' dataframes, or</li> <li>3. a dataframe containing (normalized) fluorescence values (if a time matrix or density dataframe is provided as separate argument).</li> </ol>
time	(optional) A matrix containing time values for each sample.
density	(optional) A dataframe containing density values for each sample and sample identifiers in the first three columns.
control	A fl.control object created with <a href="#">fl.control</a> , defining relevant fitting options.
...	Further arguments passed to the shiny app.

### Value

An flFit object that contains all fluorescence fitting results, compatible with various plotting functions of the QurvE package.

raw.x	Raw x matrix passed to the function as time (for x_type = 'time') or density (for x_type = 'density').
raw.fl	Raw density dataframe passed to the function as data.
flTable	Table with fluorescence parameters and related statistics for each fluorescence curve evaluation performed by the function. This table, which is also returned by the generic <code>summary.flFit</code> method applied to a flFit object, is used as an input for <a href="#">fl.drFit</a> .
flFittedLinear	List of all flFitLinear objects, generated by the call of <a href="#">flFitLinear</a> . Note: access to each object in the list via double brace: <code>flFittedLinear[[#n]]</code> .
flFittedSplines	List of all flFitSpline objects, generated by the call of <a href="#">flFitSpline</a> . Note: access to each object via double brace: <code>flFittedSplines[[#n]]</code> .
flBootSplines	List of all flBootSpline objects, generated by the call of <a href="#">flBootSpline</a> . Note: access to each object via double brace: <code>flFittedSplines[[#n]]</code> .
control	Object of class fl.control containing list of options passed to the function as control.



An flFit object that contains all fluorescence fitting results, compatible with various plotting functions of the *QuvE* package.

raw.x	Raw x dataframe passed to the function as time or density. If both arguments are used, x will be defined by 'x_type' in the <code>fl.control</code> object.
raw.fl	Raw fluorescence dataframe passed to the function as fl_data.
flTable	Table with fitted parameters and related statistics for each curve evaluation performed by the function. This table, which is also returned by the generic <code>summary.flFit</code> method applied to a flFit object, is used as an input for <code>growth.drFit</code> or <code>fl.drFit</code> .
flFittedLinear	List of all flFitLinear objects, generated by the call of <code>flFitLinear</code> . Note: access to each object in the list via double brace: <code>flFittedLinear[[#n]]</code> .
flFittedSplines	List of all flFitSpline objects, generated by the call of <code>flFitSpline</code> . Note: access to each object via double brace: <code>flFittedSplines[[#n]]</code> .
flBootSplines	List of all flBootSpline objects, generated by the call of <code>flBootSpline</code> . Note: access to each object via double brace: <code>flFittedSplines[[#n]]</code> .
control	Object of class <code>fl.control</code> containing list of options passed to the function as control.

### See Also

Other workflows: `growth.gcFit()`, `growth.workflow()`

Other fluorescence fitting functions: `flBootSpline()`, `flFitSpline()`

Other dose-response analysis functions: `growth.drBootSpline()`, `growth.drFitSpline()`, `growth.gcFit()`, `growth.workflow()`

---

flFitLinear

*Data fit via a heuristic linear method*


---

### Description

Determine maximum slopes from using a heuristic approach similar to the “growth rates made easy”-method of Hall et al. (2013).

### Usage

```
flFitLinear(
  time = NULL,
  density = NULL,
  fl_data,
  ID = "undefined",
  quota = 0.95,
  control = fl.control(x_type = c("density", "time"), log.x.lin = FALSE, log.y.lin =
    FALSE, t0 = 0, min.density = NA, lin.h = NULL, lin.R2 = 0.98, lin.RSD = 0.05, lin.dY
    = 0.05, biphasic = FALSE)
)
```

**Arguments**

time	Vector of the independent time variable (if <code>x_type = "time"</code> in control object).
density	Vector of the independent time density (if <code>x_type = "density"</code> in control object).
fl_data	Vector of the dependent fluorescence variable.
ID	(Character) The name of the analyzed sample.
quota	(Numeric, between 0 and 1) Define what fraction of <code>max_slope</code> the slope of regression windows adjacent to the window with highest slope should have to be included in the overall linear fit.
control	A <code>fl.control</code> object created with <code>fl.control</code> , defining relevant fitting options.

**Value**

A `gcFitLinear` object with parameters of the fit. The lag time is estimated as the intersection between the fit and the horizontal line with  $y = y_0$ , where  $y_0$  is the first value of the dependent variable. Use `plot.gcFitSpline` to visualize the linear fit.

raw.x	Filtered x values used for the spline fit.
raw.fl	Filtered fluorescence values used for the spline fit.
filt.x	Filtered x values.
filt.fl	Filtered fluorescence values.
ID	(Character) Identifies the tested sample.
FUN	Linear <i>function</i> used for plotting the tangent at mumax.
fit	lm object; result of the final call of <code>lm</code> to perform the linear regression.
par	List of determined fluorescence parameters.

- `y0`: Minimum fluorescence value considered for the heuristic linear method.
- `dY`: Difference in maximum fluorescence and minimum fluorescence
- `A`: Maximum fluorescence
- `y0_lm`: Intersection of the linear fit with the abscissa.
- `max_slope`: Maximum slope of the linear fit.
- `tD`: Doubling time.
- `slope.se`: Standard error of the maximum slope.
- `lag`: Lag X.
- `x.max_start`: X value of the first data point within the window used for the linear regression.
- `x.max_end`: X value of the last data point within the window used for the linear regression.
- `x.turn`: For biphasic: X at the inflection point that separates two phases.
- `max.slope2`: For biphasic: Slope of the second phase.
- `tD2`: Doubling time of the second phase.
- `y0_lm2`: For biphasic: Intersection of the linear fit of the second phase with the abscissa.
- `lag2`: For biphasic: Lag time determined for the second phase..
- `x.max2_start`: For biphasic: X value of the first data point within the window used for the linear regression of the second phase.

- `x.max2_end`: For biphasic: X value of the last data point within the window used for the linear regression of the second phase.

<code>ndx</code>	Index of data points used for the linear regression.
<code>ndx2</code>	Index of data points used for the linear regression for the second phase.
<code>control</code>	Object of class <code>grofit.control</code> containing list of options passed to the function as control.
<code>rsquared</code>	$R^2$ of the linear regression.
<code>rsquared2</code>	$R^2$ of the linear regression for the second phase.
<code>fitFlag</code>	(Logical) Indicates whether linear regression was successfully performed on the data.
<code>fitFlag2</code>	(Logical) Indicates whether a second phase was identified.
<code>reliable</code>	(Logical) Indicates whether the performed fit is reliable (to be set manually).

## References

Hall, BG., Acar, H, Nandipati, A and Barlow, M (2014) Growth Rates Made Easy. *Mol. Biol. Evol.* 31: 232-38, DOI: 10.1093/molbev/mst187

Petzoldt T (2022). *growthrates: Estimate Growth Rates from Experimental Data*. R package version 0.8.3, <https://CRAN.R-project.org/package=growthrates>.

---

flFitSpline	<i>Perform a smooth spline fit on fluorescence data</i>
-------------	---

---

## Description

flFitSpline performs a smooth spline fit on the dataset and determines the greatest slope as the global maximum in the first derivative of the spline.

## Usage

```
flFitSpline(
  time = NULL,
  density = NULL,
  fl_data,
  ID = "undefined",
  control = fl.control(biphasic = FALSE, x_type = c("density", "time"), log.x.spline =
    FALSE, log.y.spline = FALSE, smooth.fl = 0.75, t0 = 0, min.density = NA)
)
```

## Arguments

<code>time</code>	Vector of the independent variable: time (if <code>x_type = 'time'</code> in <code>fl.control</code> object).
<code>density</code>	Vector of the independent variable: density (if <code>x_type = 'density'</code> in <code>fl.control</code> object).
<code>fl_data</code>	Vector of dependent variable: fluorescence.
<code>ID</code>	(Character) The name of the analyzed sample.

control	A fl.control object created with <code>fl.control</code> , defining relevant fitting options.
biphasic	(Logical) Shall <code>flFitLinear</code> and <code>flFitSpline</code> try to extract fluorescence parameters for two different phases (as observed with, e.g., regulator-promoter systems with varying response in different growth stages) (TRUE) or not (FALSE)?
x_type	(Character) Which data type shall be used as independent variable? Options are 'density' and 'time'.
log.x.spline	(Logical) Indicates whether $\ln(x+1)$ should be applied to the independent variable for <i>spline</i> fits. Default: FALSE.
log.y.spline	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the fluorescence data for <i>spline</i> fits. Default: FALSE
smooth.fl	(Numeric) Parameter describing the smoothness of the spline fit; usually (not necessary) within (0;1]. <code>smooth.gc=NULL</code> causes the program to query an optimal value via cross validation techniques. Especially for datasets with few data points the option NULL might cause a too small smoothing parameter. This can result a too tight fit that is susceptible to measurement errors (thus overestimating slopes) or produce an error in <code>smooth.spline</code> or lead to overfitting. The usage of a fixed value is recommended for reproducible results across samples. See <code>smooth.spline</code> for further details. Default: 0.55
t0	(Numeric) Minimum time value considered for linear and spline fits.
min.density	(Numeric) Indicate whether only values above a certain threshold should be considered for linear regressions or spline fits.

## Details

If `biphasic = TRUE`, the following steps are performed to define a second phase:

1. Determine local minima within the first derivative of the smooth spline fit.
2. Remove the 'peak' containing the highest value of the first derivative (i.e.,  $mu_{max}$ ) that is flanked by two local minima.
3. Repeat the smooth spline fit and identification of maximum slope for later time values than the local minimum after  $mu_{max}$ .
4. Repeat the smooth spline fit and identification of maximum slope for earlier time values than the local minimum before  $mu_{max}$ .
5. Choose the greater of the two independently determined slopes as  $mu_{max}^2$ .

## Value

A `flFitSpline` object. The lag time is estimated as the intersection between the tangent at the maximum slope and the horizontal line with  $y = y_0$ , where  $y_0$  is the first value of the dependent variable. Use `plot.flFitSpline` to visualize the spline fit and derivative over time.

x.in	Raw x values provided to the function as time or density.
fl.in	Raw fluorescence data provided to the function as <code>fl_data</code> .
raw.x	Filtered x values used for the spline fit.
raw.fl	Filtered fluorescence values used for the spline fit.
ID	(Character) Identifies the tested sample.
fit.x	Fitted x values.

fit.fl	Fitted fluorescence values.
parameters	List of determined parameters. <ul style="list-style-type: none"> <li>• A: Maximum fluorescence.</li> <li>• dY: Difference in maximum fluorescence and minimum fluorescence.</li> <li>• max_slope: Maximum slope of fluorescence-vs.-x data (i.e., maximum in first derivative of the spline).</li> <li>• x.max: Time at the maximum slope.</li> <li>• lambda: Lag time.</li> <li>• b.tangent: Intersection of the tangent at the maximum slope with the abscissa.</li> <li>• max_slope2: For biphasic course of fluorescence: Maximum slope of fluorescence-vs.-x data of the second phase.</li> <li>• lambda2: For biphasic course of fluorescence: Lag time determined for the second phase.</li> <li>• x.max2: For biphasic course of fluorescence: Time at the maximum slope of the second phase.</li> <li>• b.tangent2: For biphasic course of fluorescence: Intersection of the tangent at the maximum slope of the second phase with the abscissa.</li> <li>• integral: Area under the curve of the spline fit.</li> </ul>
spline	smooth.spline object generated by the <a href="#">smooth.spline</a> function.
spline.deriv1	list of time ('x') and density ('y') values describing the first derivative of the spline fit.
reliable	(Logical) Indicates whether the performed fit is reliable (to be set manually).
fitFlag	(Logical) Indicates whether a spline fit was successfully performed on the data.
fitFlag2	(Logical) Indicates whether a second phase was identified.
control	Object of class fl.control containing list of options passed to the function as control.

### See Also

Other fluorescence fitting functions: [flBootSpline\(\)](#), [flFit\(\)](#)

---

grofit.param	<i>Internal function called within <a href="#">growth.gcFitModel</a>.</i>
--------------	---

---

### Description

Internal function called within [growth.gcFitModel](#).

### Usage

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

### Arguments

time	Vector of the independent variable (usually time).
data	Vector of dependent variable (usually density values).
gcID	(Character) The name of the analyzed sample.
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

growth.control

*Create a grofit.control object.***Description**

A grofit.control object is required to perform various computations on growth data stored within grodata objects (created with [read\\_data](#) or [parse\\_data](#)). A grofit.control object is created automatically as part of [growth.workflow](#).

**Usage**

```
growth.control(
  neg.nan.act = FALSE,
  clean.bootstrap = TRUE,
  suppress.messages = FALSE,
  fit.opt = c("a"),
  t0 = 0,
  tmax = NA,
  min.density = NA,
  max.density = NA,
  log.x.gc = FALSE,
  log.y.lin = TRUE,
  log.y.spline = TRUE,
  log.y.model = TRUE,
  lin.h = NULL,
  lin.R2 = 0.97,
  lin.RSD = 0.1,
  lin.dY = 0.05,
  biphasic = FALSE,
  interactive = FALSE,
  nboot.gc = 0,
  smooth.gc = 0.55,
  model.type = c("logistic", "richards", "gompertz", "gompertz.exp", "huang", "baranyi"),
  dr.method = c("model", "spline"),
  dr.model = c("gammadr", "multi2", "LL.2", "LL.3", "LL.4", "LL.5", "W1.2", "W1.3",
    "W1.4", "W2.2", "W2.3", "W2.4", "LL.3u", "LL2.2", "LL2.3", "LL2.3u", "LL2.4",
    "LL2.5", "AR.2", "AR.3"),
  dr.have.atleast = 6,
  dr.parameter = c("mu.linfit", "lambda.linfit", "dY.linfit", "A.linfit", "mu.spline",
    "lambda.spline", "dY.spline", "A.spline", "mu.model", "lambda.model",
    "dY.orig.model", "A.orig.model"),
  smooth.dr = NULL,
  log.x.dr = FALSE,
  log.y.dr = FALSE,
  nboot.dr = 0,
  growth.thresh = 1.5
)
```

**Arguments**

neg.nan.act      (Logical) Indicates whether the program should stop when negative growth values or NA values appear (TRUE). Otherwise, the program removes these 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 whether 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 throughput data. Note: warnings are still displayed. Default: FALSE.
fit.opt	(Character or character vector) Indicates whether the program should perform a linear regression ("l"), model fit ("m"), spline fit ("s"), or all ("a"). Combinations can be freely chosen by providing a character vector, e.g. <code>fit.opt = c("l", "s")</code> Default: <code>fit.opt = c("l", "s")</code> .
t0	(Numeric) Minimum time value considered for linear and spline fits.
tmax	(Numeric) Maximum time value considered for linear and spline fits.
min.density	(Numeric) Indicate whether only density values above a certain threshold should be considered for linear regressions or spline fits.
max.density	(Numeric) Indicate whether only density values below a certain threshold should be considered for linear regressions or spline fits.
log.x.gc	(Logical) Indicates whether $\ln(x+1)$ should be applied to the time data for <i>linear</i> and <i>spline</i> fits. Default: FALSE.
log.y.lin	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>linear</i> fits. Default: TRUE
log.y.spline	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>spline</i> fits. Default: TRUE
log.y.model	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>model</i> fits. Default: TRUE
lin.h	(Numeric) Manually define the size of the sliding window used in <a href="#">growth.gcFitLinear</a> . If NULL, h is calculated for each samples based on the number of measurements in the growth phase of the plot.
lin.R2	(Numeric) $R^2$ threshold for <a href="#">growth.gcFitLinear</a>
lin.RSD	(Numeric) Relative standard deviation (RSD) threshold for the calculated slope in <a href="#">growth.gcFitLinear</a>
lin.dY	(Numeric) Threshold for the minimum fraction of density increase a linear regression window should cover. Default: 0.05 (5%).
biphasic	(Logical) Shall <a href="#">growth.gcFitLinear</a> and <a href="#">growth.gcFitSpline</a> try to extract growth parameters for two different growth phases (as observed with, e.g., diauxic shifts) (TRUE) or not (FALSE)?
interactive	(Logical) Controls whether the fit of each growth curve and method is controlled manually by the user. If TRUE, each fit is visualized in the <i>Plots</i> pane and the user can adjust fitting parameters and confirm the reliability of each fit per sample. Default: TRUE.
nboot.gc	(Numeric) Number of bootstrap samples used for nonparametric growth curve fitting with <a href="#">growth.gcBootSpline</a> . Use <code>nboot.gc = 0</code> to disable the bootstrap. Default: 0

<code>smooth.gc</code>	(Numeric) Parameter describing the smoothness of the spline fit; usually (not necessary) within (0;1]. <code>smooth.gc=NULL</code> causes the program to query an optimal value via cross validation techniques. Especially for datasets with few data points the option <code>NULL</code> might cause a too small smoothing parameter. This can result a too tight fit that is susceptible to measurement errors (thus overestimating growth rates) or produce an error in <a href="#">smooth.spline</a> or lead to overfitting. The usage of a fixed value is recommended for reproducible results across samples. See <a href="#">smooth.spline</a> for further details. Default: 0.55
<code>model.type</code>	(Character) Vector providing the names of the parametric models which should be fitted to the data. Default: <code>c("logistic", "richards", "gompertz", "gompertz.exp", "huang", "baranyi")</code> .
<code>dr.method</code>	(Character) Define the method used to perform a dose-response analysis: smooth spline fit ( <code>"spline"</code> ) or model fitting ( <code>"model"</code> ).
<code>dr.model</code>	(Character) Provide a list of models from the R package 'drc' to include in the dose-response analysis (if <code>dr.method = "model"</code> ). If more than one model is provided, the best-fitting model will be chosen based on the Akaike Information Criterion.
<code>dr.have.atleast</code>	(Numeric) 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>dr.parameter</code>	(Character or numeric) The response parameter in the output table to be used for creating a dose response curve. See <a href="#">growth.drFit</a> for further details. Default: <code>"mu.linfit"</code> , which represents the maximum slope of the linear regression. Typical options include: <code>"mu.linfit"</code> , <code>"lambda.linfit"</code> , <code>"dY.linfit"</code> , <code>"mu.spline"</code> , <code>"dY.spline"</code> , <code>"mu.model"</code> , and <code>"A.model"</code> .
<code>smooth.dr</code>	(Numeric) Smoothing parameter used in the spline fit by <code>smooth.spline</code> during dose response curve estimation. Usually (not necessary) in (0; 1]. See <a href="#">smooth.spline</a> for further details. Default: <code>NULL</code> .
<code>log.x.dr</code>	(Logical) Indicates whether $\ln(x+1)$ should be applied to the concentration data of the dose response curves. Default: <code>FALSE</code> .
<code>log.y.dr</code>	(Logical) Indicates whether $\ln(y+1)$ should be applied to the response data of the dose response curves. Default: <code>FALSE</code> .
<code>nboot.dr</code>	(Numeric) Defines the number of bootstrap samples for EC50 estimation. Use <code>nboot.dr = 0</code> to disable bootstrapping. Default: 0.
<code>growth.thresh</code>	(Numeric) Define a threshold for growth. Only if any density value in a sample is greater than <code>growth.thresh</code> (default: 1.5) times the start density, further computations are performed. Else, a message is returned.

## Value

Generates a list with all arguments described above as entries.

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07



---

growth.drBootSpline	<i>Perform a smooth spline fit on response vs. concentration data of a single sample</i>
---------------------	--

---

## Description

growth.drBootSpline resamples the values in a dataset with replacement and performs a spline fit for each bootstrap sample to determine the EC50.

## Usage

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

## Arguments

conc	Vector of concentration values.
test	Vector of response parameter values of the same length as conc.
drID	(Character) The name of the analyzed sample.
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

## Value

An object of class drBootSpline containing a distribution of growth parameters and a drFitSpline object for each bootstrap sample. Use [plot.drBootSpline](#) to visualize all bootstrapping splines as well as the distribution of EC50.

raw.conc	Raw data provided to the function as conc.
raw.test	Raw data for the response parameter provided to the function as test.
drID	(Character) Identifies the tested condition.
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 drFitSpline objects generated by the call of <a href="#">growth.drFitSpline</a> .
ec50.boot	Vector of estimated EC50 values from each bootstrap entry.
ec50y.boot	Vector of estimated response at EC50 values from each bootstrap entry.
BootFlag	(Logical) Indicates the success of the bootstrapping operation.
control	Object of class grofit.control containing list of options passed to the function as control.

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

## See Also

Other dose-response analysis functions: [f1Fit\(\)](#), [growth.drFitSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

growth.drFit

*Perform smooth spline fits on response vs. concentration data***Description**

growth.drFit serves to determine dose-response curves on every condition in a dataset. The response parameter can be chosen from every physiological parameter in a gcTable table which is obtained via [growth.gcFit](#). growth.drFit calls the functions [growth.drFitSpline](#) and [growth.drBootSpline](#) to generate a table with estimates for EC50 and respecting statistics.

**Usage**

```
growth.drFit(gcTable, control = growth.control())
```

**Arguments**

gcTable	A dataframe containing the data for the dose-response curve estimation. Such table of class gcTable can be obtained by running <a href="#">growth.gcFit</a> .
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

**Value**

An object of class drFit.

raw.data	Data that passed to the function as gcTable.
drTable	Dataframe containing condition identifiers, fit options, and results of the dose-response analysis.
drBootSplines	List of all drBootSpline objects generated by the call of <a href="#">growth.drBootSpline</a> for each distinct experiment.
drFittedSplines	List of all drFitSpline objects generated by the call of <a href="#">growth.drFitSpline</a> for each distinct experiment.
control	Object of class grofit.control containing list of options passed to the function as control.

**References**

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

**See Also**

Other growth fitting functions: [growth.gcBootSpline\(\)](#), [growth.gcFitLinear\(\)](#), [growth.gcFitModel\(\)](#), [growth.gcFitSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

growth.drFitModel	<i>Fit various models to response vs. concentration data of a single sample to determine the EC50.</i>
-------------------	--

---

**Description**

Fit various models to response vs. concentration data of a single sample to determine the EC50.

**Usage**

```
growth.drFitModel(conc, test, drID = "undefined", control = growth.control())
```

**Arguments**

conc	Vector of concentration values.
test	Vector of response parameter values of the same length as conc.
drID	(Character) The name of the analyzed condition
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

**Value**

A drFitModel object.

**References**

Christian Ritz, Florent Baty, Jens C. Streibig, Daniel Gerhard (2015). *Dose-Response Analysis Using R*. PLoS ONE 10(12): e0146021. DOI: 10.1371/journal.pone.0146021

---

growth.drFitSpline	<i>Perform a smooth spline fit on response vs. concentration data of a single sample to determine the EC50.</i>
--------------------	---

---

**Description**

growth.drFitSpline performs a smooth spline fit determines the EC50 as the concentration at the half-maximum value of the response parameter of the spline.

**Usage**

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

**Arguments**

conc	Vector of concentration values.
test	Vector of response parameter values of the same length as conc.
drID	(Character) The name of the analyzed condition
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

## Details

During the spline fit with [smooth.spline](#), higher weights are assigned to data points with a concentration value of 0, as well as to x-y pairs with a response parameter value of 0 and pairs at concentration values before zero-response parameter values.

## Value

A drFitSpline object.

raw.conc	Raw data provided to the function as conc.
raw.test	Raw data for the response parameter provided to the function as test.
drID	(Character) Identifies the tested condition
fit.conc	Fitted concentration values.
fit.test	Fitted response values.
spline	smooth.spline object generated by the <a href="#">smooth.spline</a> function.
spline.low	x and y values of <a href="#">lowess</a> spline fit on raw data. Used to call <a href="#">smooth.spline</a> .
parameters	List of parameters estimated from dose response curve fit. <ul style="list-style-type: none"> <li>• EC50: Concentration at half-maximal response.</li> <li>• yEC50: Response value related to EC50.</li> <li>• EC50.orig: EC50 value in original scale, if a transformation was applied.</li> <li>• yEC50.orig: Response value for EC50 in original scale, if a transformation was applied.</li> </ul>
fitFlag	(Logical) Indicates whether a spline could fitted successfully to data.
reliable	(Logical) Indicates whether the performed fit is reliable (to be set manually).
control	Object of class grofit.control containing list of options passed to the function as control.

Use [plot.drFitSpline](#) to visualize the spline fit.

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

## See Also

Other dose-response analysis functions: [flFit\(\)](#), [growth.drBootSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

growth.gcBootSpline	<i>Perform a bootstrap on density vs. time data followed by spline fits for each resample</i>
---------------------	---

---

### Description

growth.gcBootSpline resamples the density-time value pairs in a dataset with replacement and performs a spline fit for each bootstrap sample.

### Usage

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

### Arguments

time	Vector of the independent variable (usually: time).
data	Vector of dependent variable (usually: density values).
gcID	(Character) The name of the analyzed sample.
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

### Value

A gcBootSpline object containing a distribution of growth parameters and a gcFitSpline object for each bootstrap sample. Use [plot.gcBootSpline](#) to visualize all bootstrapping splines as well as the distribution of physiological parameters.

raw.time	Raw time values provided to the function as time.
raw.data	Raw density data provided to the function as data.
gcID	(Character) Identifies the tested sample.
boot.time	Table of time values per column, resulting from each spline fit of the bootstrap.
boot.data	Table of density values per column, resulting from each spline fit of the bootstrap.
boot.gcSpline	List of gcFitSpline object, created by <a href="#">growth.gcFitSpline</a> for each resample of the bootstrap.
lambda	Vector of estimated lambda (lag time) values from each bootstrap entry.
mu	Vector of estimated mu (maximum growth rate) values from each bootstrap entry.
A	Vector of estimated A (maximum density) values from each bootstrap entry.
integral	Vector of estimated integral values from each bootstrap entry.
bootFlag	(Logical) Indicates the success of the bootstrapping operation.
control	Object of class grofit.control containing list of options passed to the function as control.

### References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

**See Also**

Other growth fitting functions: [growth.drFit\(\)](#), [growth.gcFitLinear\(\)](#), [growth.gcFitModel\(\)](#), [growth.gcFitSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

growth.gcFit	<i>Perform a growth curve analysis on all samples in the provided dataset.</i>
--------------	--

---

**Description**

growth.gcFit performs all computational growth fitting operations based on the user input.

**Usage**

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

**Arguments**

time	(optional) A matrix containing time values for each sample.
data	Either... <ol style="list-style-type: none"> <li>1. a grodata object created with <a href="#">read_data</a> or <a href="#">parse_data</a>,</li> <li>2. a list containing a 'time' matrix as well as 'density' and, if appropriate, a 'fluorescence' dataframes, or</li> <li>3. a dataframe containing density values (if a time matrix is provided as separate argument).</li> </ol>
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.
...	Further arguments passed to the shiny app.

**Value**

A gcFit object that contains all growth fitting results, compatible with various plotting functions of the QurvE package.

raw.time	Raw time matrix passed to the function as time.
raw.data	Raw density dataframe passed to the function as data.
gcTable	Table with growth parameters and related statistics for each growth curve evaluation performed by the function. This table, which is also returned by the generic <code>summary.gcFit</code> method applied to a gcFit object, is used as an input for <a href="#">growth.drFit</a> .
gcFittedLinear	List of all gcFitLinear objects, generated by the call of <a href="#">growth.gcFitLinear</a> . Note: access to each object in the list via double brace: <code>gcFittedLinear[[#n]]</code> .
gcFittedModels	List of all gcFitModel objects, generated by the call of <a href="#">growth.gcFitModel</a> . Note: access to each object in the list via double brace: <code>gcFittedModels[[#n]]</code> .
gcFittedSplines	List of all gcFitSpline objects, generated by the call of <a href="#">growth.gcFitSpline</a> . Note: access to each object via double brace: <code>gcFittedSplines[[#n]]</code> .
gcBootSplines	List of all gcBootSpline objects, generated by the call of <a href="#">growth.gcBootSpline</a> . Note: access to each object via double brace: <code>gcFittedSplines[[#n]]</code> .
control	Object of class <code>grofit.control</code> containing list of options passed to the function as control.

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

## See Also

Other workflows: `flFit()`, `growth.workflow()`

Other growth fitting functions: `growth.drFit()`, `growth.gcBootSpline()`, `growth.gcFitLinear()`, `growth.gcFitModel()`, `growth.gcFitSpline()`, `growth.workflow()`

Other dose-response analysis functions: `flFit()`, `growth.drBootSpline()`, `growth.drFitSpline()`, `growth.workflow()`

---

growth.gcFitLinear	<i>Fit an exponential growth model with a heuristic linear method</i>
--------------------	---

---

## Description

Determine maximum growth rates from the log-linear part of a growth curve using a heuristic approach similar to the “growth rates made easy”-method of Hall et al. (2013).

## Usage

```
growth.gcFitLinear(
  time,
  data,
  gcID = "undefined",
  quota = 0.95,
  control = growth.control(t0 = 0, tmax = NA, log.x.gc = FALSE, log.y.lin = TRUE,
    min.density = NA, max.density = NA, lin.h = NULL, lin.R2 = 0.97, lin.RSD = 0.1,
    lin.dY = 0.05, biphasic = FALSE)
)
```

## Arguments

time	Vector of the independent variable (usually: time).
data	Vector of dependent variable (usually: density values).
gcID	(Character) The name of the analyzed sample.
quota	(Numeric, between 0 and 1) Define what fraction of $\mu_{max}$ the slope of regression windows adjacent to the window with highest slope should have to be included in the overall linear fit.
control	A <code>grofit.control</code> object created with <code>growth.control</code> , defining relevant fitting options.
log.x.gc	(Logical) Indicates whether $\ln(x+1)$ should be applied to the time data for <i>linear</i> and <i>spline</i> fits. Default: FALSE.
log.y.lin	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>linear</i> fits. Default: TRUE

min.density	(Numeric) Indicate whether only density values above a certain threshold should be considered for linear regressions.
max.density	(Numeric) Indicate whether only density values below a certain threshold should be considered for linear regressions.
t0	(Numeric) Minimum time value considered for linear and spline fits.
tmax	(Numeric) Minimum time value considered for linear and spline fits.
lin.h	(Numeric) Manually define the size of the sliding window . If NULL, h is calculated for each samples based on the number of measurements in the growth phase of the plot.
lin.R2	(Numeric) $R^2$ threshold for <a href="#">growth.gcFitLinear</a>
lin.RSD	(Numeric) Relative standard deviation (RSD) threshold for calculated slope in <a href="#">growth.gcFitLinear</a>
lin.dY	(Numeric) Enter the minimum percentage of density increase that a linear regression should cover.
biphasic	(Logical) Shall <a href="#">growth.gcFitLinear</a> try to extract growth parameters for two different growth phases (as observed with, e.g., diauxic shifts) (TRUE) or not (FALSE)?

## Details

The algorithm works as follows:

1. Fit linear regressions (Theil-Sen estimator) to all subsets of  $h$  consecutive, log-transformed data points (sliding window of size  $h$ ). If for example  $h = 5$ , fit a linear regression to points 1 ... 5, 2 ... 6, 3 ... 7 and so on.
2. Find the subset with the highest slope  $\mu_{max}$ . Do the  $R^2$  and relative standard deviation (RSD) values of the regression meet the in `lin.R2` and `lin.RSD` defined thresholds and do the data points within the regression window account for a fraction of at least `lin.dY` of the total density increase? If not, evaluate the subset with the second highest slope, and so on.
3. Include also the data points of adjacent subsets that have a slope of at least  $quota \cdot \mu_{max}$ , e.g., all regression windows that have at least 95% of the maximum slope.
4. Fit a new linear model to the extended data window identified in step 3.

If `biphasic = TRUE`, the following steps are performed to define a second growth phase:

1. Perform a smooth spline fit on the data with a smoothing factor of 0.5.
2. Calculate the second derivative of the spline fit and perform a smooth spline fit of the derivative with a smoothing factor of 0.4.
3. Determine local maxima and minima in the second derivative.
4. Find the local minimum following  $\mu_{max}$  and repeat the heuristic linear method for later time values.
5. Find the local maximum before  $\mu_{max}$  and repeat the heuristic linear method for earlier time values.
6. Choose the greater of the two independently determined slopes as  $\mu_{max}2$ .



**Value**

A gcFitLinear object with parameters of the fit. The lag time is estimated as the intersection between the fit and the horizontal line with  $y = y_0$ , where  $y_0$  is the first value of the dependent variable. Use [plot.gcFitSpline](#) to visualize the linear fit.

raw.time	Raw time values provided to the function as time.
raw.data	Raw density data provided to the function as data.
filt.time	Filtered time values used for the heuristic linear method.
filt.data	Filtered density values.
log.data	Log-transformed, filtered density values used for the heuristic linear method.
gcID	(Character) Identifies the tested sample.
FUN	Linear <i>function</i> used for plotting the tangent at mumax.
fit	lm object; result of the final call of <a href="#">lm</a> to perform the linear regression.
par	List of determined growth parameters.
<ul style="list-style-type: none"> <li>• <math>y_0</math>: Minimum density value considered for the heuristic linear method.</li> <li>• dY: Difference in maximum density and minimum density.</li> <li>• A: Maximum density.</li> <li>• <math>y_0\_lm</math>: Intersection of the linear fit with the abscissa.</li> <li>• mumax: Maximum growth rate (i.e., slope of the linear fit).</li> <li>• tD: Doubling time.</li> <li>• mu.se: Standard error of the maximum growth rate.</li> <li>• lag: Lag time.</li> <li>• tmax_start: Time value of the first data point within the window used for the linear regression.</li> <li>• tmax_end: Time value of the last data point within the window used for the linear regression.</li> <li>• t_turn: For biphasic growth: Time of the inflection point that separates two growth phases.</li> <li>• mumax2: For biphasic growth: Growth rate of the second growth phase.</li> <li>• tD2: Doubling time of the second growth phase.</li> <li>• <math>y_0\_lm2</math>: For biphasic growth: Intersection of the linear fit of the second growth phase with the abscissa.</li> <li>• lag2: For biphasic growth: Lag time determined for the second growth phase..</li> <li>• tmax2_start: For biphasic growth: Time value of the first data point within the window used for the linear regression of the second growth phase.</li> <li>• tmax2_end: For biphasic growth: Time value of the last data point within the window used for the linear regression of the second growth phase.</li> </ul>	
ndx	Index of data points used for the linear regression.
ndx2	Index of data points used for the linear regression for the second growth phase.
control	Object of class grofit.control containing list of options passed to the function as control.
rsquared	$R^2$ of the linear regression.
rsquared2	$R^2$ of the linear regression for the second growth phase.
fitFlag	(Logical) Indicates whether linear regression was successfully performed on the data.
fitFlag2	(Logical) Indicates whether a second growth phase was identified.
reliable	(Logical) Indicates whether the performed fit is reliable (to be set manually).

## References

Hall, BG., Acar, H, Nandipati, A and Barlow, M (2014) Growth Rates Made Easy. *Mol. Biol. Evol.* 31: 232-38, DOI: 10.1093/molbev/mst187

Petzoldt T (2022). growthrates: Estimate Growth Rates from Experimental Data. R package version 0.8.3, <https://CRAN.R-project.org/package=growthrates>.

Theil, H.(1992). A rank-invariant method of linear and polynomial regression analysis. In: Henri Theil's contributions to economics and econometrics. Springer, pp. 345–381. DOI: 10.1007/978-94-011-2546-8\_20

## See Also

Other growth fitting functions: [growth.drFit\(\)](#), [growth.gcBootSpline\(\)](#), [growth.gcFitModel\(\)](#), [growth.gcFitSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

growth.gcFitModel	<i>Fit nonlinear growth models to density data</i>
-------------------	--

---

## Description

growth.gcFitModel determines a parametric growth model that best describes the data.

## Usage

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

## Arguments

time	Vector of the independent variable (usually time).
data	Vector of dependent variable (usually density values).
gcID	(Character) The name of the analyzed sample.
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.

## Value

A gcFitModel object that contains physiological parameters and information about the best fit. Use [plot.gcFitModel](#) to visualize the parametric fit and growth equation.

raw.time	Raw time values provided to the function as time.
raw.data	Raw density data provided to the function as data.
gcID	(Character) Identifies the tested sample.
fit.time	Fitted time values.
fit.data	Fitted density values.
parameters	List of determined growth parameters.

- A: Maximum density.
- dY: Difference in maximum density and minimum density of the fitted model.
- mu: Maximum growth rate (i.e., maximum in first derivative of the spline).

- `lambda`: Lag time.
- `b.tangent`: Intersection of the tangent at the maximum growth rate with the abscissa.
- `fitpar`: For some models: list of additional parameters used in the equations describing the growth curve.
- `integral`: Area under the curve of the parametric fit.

<code>model</code>	(Character) The model that obtained the fit with the lowest AIC, determined by <a href="#">AIC</a> .
<code>nls</code>	<code>nls</code> object for the chosen model generated by the <a href="#">nls</a> function.
<code>reliable</code>	(Logical) Indicates whether the performed fit is reliable (to be set manually).
<code>fitFlag</code>	(Logical) Indicates whether a parametric model was successfully fitted on the data.
<code>control</code>	Object of class <code>grofit.control</code> containing list of options passed to the function as <code>control</code> .

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

## See Also

Other growth fitting functions: [growth.drFit\(\)](#), [growth.gcBootSpline\(\)](#), [growth.gcFitLinear\(\)](#), [growth.gcFitSpline\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

<code>growth.gcFitSpline</code>	<i>Perform a smooth spline fit on growth data</i>
---------------------------------	---

---

## Description

`growth.gcFitSpline` performs a smooth spline fit on the dataset and determines the highest growth rate as the global maximum in the first derivative of the spline.

## Usage

```
growth.gcFitSpline(
  time,
  data,
  gcID = "undefined",
  control = growth.control(biphasic = FALSE)
)
```

**Arguments**

time	Vector of the independent variable (usually time).
data	Vector of dependent variable (usually: density values).
gcID	(Character) The name of the analyzed sample.
control	A grofit.control object created with <a href="#">growth.control</a> , defining relevant fitting options.
biphasic	(Logical) Shall growth.gcFitSpline try to extract growth parameters for two different growth phases (as observed with, e.g., diauxic shifts) (TRUE) or not (FALSE)?

**Details**

If biphasic = TRUE, the following steps are performed to define a second growth phase:

1. Determine local minima within the first derivative of the smooth spline fit.
2. Remove the 'peak' containing the highest value of the first derivative (i.e.,  $\mu_{max}$ ) that is flanked by two local minima.
3. Repeat the smooth spline fit and identification of maximum slope for later time values than the local minimum after  $\mu_{max}$ .
4. Repeat the smooth spline fit and identification of maximum slope for earlier time values than the local minimum before  $\mu_{max}$ .
5. Choose the greater of the two independently determined slopes as  $\mu_{max}2$ .

**Value**

A gcFitSpline object. The lag time is estimated as the intersection between the tangent at the maximum slope and the horizontal line with  $y = y_0$ , where  $y_0$  is the first value of the dependent variable. Use [plot.gcFitSpline](#) to visualize the spline fit and derivative over time.

time.in	Raw time values provided to the function as time.
data.in	Raw density data provided to the function as data.
raw.time	Filtered time values used for the spline fit.
raw.data	Filtered density values used for the spline fit.
gcID	(Character) Identifies the tested sample.
fit.time	Fitted time values.
fit.data	Fitted density values.
parameters	List of determined growth parameters.

- A: Maximum density.
- dY: Difference in maximum density and minimum density.
- mu: Maximum growth rate (i.e., maximum in first derivative of the spline).
- tD: Doubling time.
- t.max: Time at the maximum growth rate.
- lambda: Lag time.
- b.tangent: Intersection of the tangent at the maximum growth rate with the abscissa.
- mu2: For biphasic growth: Growth rate of the second growth phase.

- `tD2`: Doubling time of the second growth phase.
- `lambda2`: For biphasic growth: Lag time determined for the second growth phase.
- `t.max2`: For biphasic growth: Time at the maximum growth rate of the second growth phase.
- `b.tangent2`: For biphasic growth: Intersection of the tangent at the maximum growth rate of the second growth phase with the abscissa.
- `integral`: Area under the curve of the spline fit.

<code>spline</code>	smooth.spline object generated by the <a href="#">smooth.spline</a> function.
<code>spline.deriv1</code>	list of time ('x') and density ('y') values describing the first derivative of the spline fit.
<code>reliable</code>	(Logical) Indicates whether the performed fit is reliable (to be set manually).
<code>fitFlag</code>	(Logical) Indicates whether a spline fit was successfully performed on the data.
<code>fitFlag2</code>	(Logical) Indicates whether a second growth phase was identified.
<code>control</code>	Object of class <code>grofit.control</code> containing list of options passed to the function as control.

## References

Matthias Kahm, Guido Hasenbrink, Hella Lichtenberg-Frate, Jost Ludwig, Maik Kschischo (2010). *grofit: Fitting Biological Growth Curves with R*. Journal of Statistical Software, 33(7), 1-21. DOI: 10.18637/jss.v033.i07

## See Also

Other growth fitting functions: [growth.drFit\(\)](#), [growth.gcBootSpline\(\)](#), [growth.gcFitLinear\(\)](#), [growth.gcFitModel\(\)](#), [growth.gcFit\(\)](#), [growth.workflow\(\)](#)

---

<code>growth.report</code>	<i>Create a PDF and HTML report with results from a growth curve analysis workflow</i>
----------------------------	--

---

## Description

`growth.report` requires a `grofit` object and creates a report in PDF and HTML format that summarizes all results.

## Usage

```
growth.report(
  grofit,
  out.dir = NULL,
  out.nm = NULL,
  ec50 = FALSE,
  format = c("pdf", "html"),
  export = FALSE,
  ...
)
```

## Arguments

grofit	A grofit object created with <a href="#">growth.workflow</a> .
out.dir	(Character) The path or name of the folder in which the report files are created. If NULL, the folder will be named with a combination of "Report.growth_" and the current date and time.
out.nm	Character or NULL Define the name of the report files. If NULL, the files will be named with a combination of "GrowthReport_" and the current date and time.
ec50	(Logical) Display results of dose-response analysis (TRUE) or not (FALSE).
format	(Character) Define the file format for the report, PDF ('pdf') and/or HTML ('html'). Default: (c('pdf', 'html'))
export	(Logical) Shall all plots generated in the report be exported as individual PDF and PNG files TRUE or not FALSE?
...	Further arguments passed to create a report. Currently supported: <ul style="list-style-type: none"> <li>• mean.grp: Define groups to combine into common plots in the report based on sample identifiers. Partial matches with sample/group names are accepted. Can be "all", a string vector, or a list of string vectors. Note: The maximum number of sample groups (with unique condition/concentration indicators) is 50. If you have more than 50 groups, option "all" will produce the error ! Insufficient values in manual scale. [Number] needed but only 50 provided.</li> <li>• mean.conc: Define concentrations to combine into common plots in the report. Can be a numeric vector, or a list of numeric vectors.</li> </ul>

---

growth.workflow	<i>Run a complete growth curve analysis and dose-reponse analysis workflow.</i>
-----------------	---

---

## Description

growth.workflow runs [growth.control](#) to create a grofit.control object and then performs all computational fitting operations based on the user input. Finally, if desired, a final report is created in PDF or HTML format that summarizes all results obtained.

## Usage

```
growth.workflow(
  grodata = NULL,
  time = NULL,
  data = NULL,
  ec50 = TRUE,
  mean.grp = NA,
  mean.conc = NA,
  neg.nan.act = FALSE,
  clean.bootstrap = TRUE,
  suppress.messages = FALSE,
  fit.opt = c("a"),
  t0 = 0,
  tmax = NA,
```

```

min.density = NA,
max.density = NA,
log.x.gc = FALSE,
log.y.lin = TRUE,
log.y.spline = TRUE,
log.y.model = TRUE,
biphasic = FALSE,
lin.h = NULL,
lin.R2 = 0.97,
lin.RSD = 0.1,
lin.dY = 0.05,
interactive = FALSE,
nboot.gc = 0,
smooth.gc = 0.55,
model.type = c("logistic", "richards", "gompertz", "gompertz.exp", "huang", "baranyi"),
dr.method = c("model", "spline"),
dr.model = c("gammadr", "multi2", "LL.2", "LL.3", "LL.4", "LL.5", "W1.2", "W1.3",
  "W1.4", "W2.2", "W2.3", "W2.4", "LL.3u", "LL2.2", "LL2.3", "LL2.3u", "LL2.4",
  "LL2.5", "AR.2", "AR.3"),
growth.thresh = 1.5,
dr.have.atleast = 6,
dr.parameter = c("mu.linfit", "lambda.linfit", "dY.linfit", "A.linfit", "mu.spline",
  "lambda.spline", "dY.spline", "A.spline", "mu.model", "lambda.model",
  "dY.orig.model", "A.orig.model"),
smooth.dr = 0.1,
log.x.dr = FALSE,
log.y.dr = FALSE,
nboot.dr = 0,
report = NULL,
out.dir = NULL,
out.nm = NULL,
export.fig = FALSE,
export.res = FALSE,
...
)

```

## Arguments

grodata	A grodata object created with <a href="#">read_data</a> or <a href="#">parse_data</a> , or a list containing a 'time' matrix as well as a 'density' dataframe.
time	(optional) A matrix containing time values for each sample.
data	(optional) A dataframe containing growth data (if a time matrix is provided as separate argument).
ec50	(Logical) Perform dose-response analysis (TRUE) or not (FALSE).
mean.grp	("all", a string vector, or a list of string vectors) Define groups to combine into common plots in the final report based on sample identifiers (if report == TRUE). Partial matches with sample/group names are accepted. Note: The maximum number of sample groups (with unique condition/concentration indicators) is 50. If you have more than 50 groups, option "all" will produce the error ! Insufficient values in manual scale. [Number] needed but only 50 provided.
mean.conc	(A numeric vector, or a list of numeric vectors) Define concentrations to combine into common plots in the final report (if report == TRUE).

neg.nan.act	(Logical) Indicates whether the program should stop when negative growth values or NA values appear (TRUE). Otherwise, the program removes these 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 whether 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 high-throughput processing data. Note: warnings are still displayed. Default: FALSE.
fit.opt	(Character or character vector) Indicates whether the program should perform a linear regression ("l"), model fit ("m"), spline fit ("s"), or all ("a"). Combinations can be freely chosen by providing a character vector, e.g. fit.opt = c("l", "s") Default: fit.opt = c("l", "s").
t0	(Numeric) Minimum time value considered for linear and spline fits.
tmax	(Numeric) Maximum time value considered for linear and spline fits.
min.density	(Numeric) Indicate whether only density values above a certain threshold should be considered for linear regressions or spline fits.
max.density	(Numeric) Indicate whether only density values below a certain threshold should be considered for linear regressions or spline fits.
log.x.gc	(Logical) Indicates whether $\ln(x+1)$ should be applied to the time data for <i>linear</i> and <i>spline</i> fits. Default: FALSE.
log.y.lin	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>linear</i> fits. Default: TRUE
log.y.spline	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>spline</i> fits. Default: TRUE
log.y.model	(Logical) Indicates whether $\ln(y/y_0)$ should be applied to the growth data for <i>model</i> fits. Default: TRUE
biphasic	(Logical) Shall <a href="#">growth.gcFitLinear</a> and <a href="#">growth.gcFitSpline</a> try to extract growth parameters for two different growth phases (as observed with, e.g., diauxic shifts) (TRUE) or not (FALSE)?
lin.h	(Numeric) Manually define the size of the sliding window used in <a href="#">growth.gcFitLinear</a> . If NULL, h is calculated for each samples based on the number of measurements in the growth phase of the plot.
lin.R2	(Numeric) $R^2$ threshold for <a href="#">growth.gcFitLinear</a>
lin.RSD	(Numeric) Relative standard deviation (RSD) threshold for calculated slope in <a href="#">growth.gcFitLinear</a>
lin.dY	(Numeric) Threshold for the minimum fraction of density increase a linear regression window should cover. Default: 0.05 (5%).
interactive	(Logical) Controls whether the fit of each growth curve and method is controlled manually by the user. If TRUE, each fit is visualized in the <i>Plots</i> pane and the user can adjust fitting parameters and confirm the reliability of each fit per sample. Default: TRUE.
nboot.gc	(Numeric) Number of bootstrap samples used for nonparametric growth curve fitting with <a href="#">growth.gcBootSpline</a> . Use nboot.gc = 0 to disable the bootstrap. Default: 0



smooth.gc	(Numeric) Parameter describing the smoothness of the spline fit; usually (not necessary) within (0;1]. smooth.gc=NULL causes the program to query an optimal value via cross validation techniques. Especially for datasets with few data points the option NULL might cause a too small smoothing parameter. This can result a too tight fit that is susceptible to measurement errors (thus overestimating growth rates) or produce an error in smooth.spline or lead to an overestimation. The usage of a fixed value is recommended for reproducible results across samples. See ?smooth.spline for further details. Default: 0.55
model.type	(Character) Vector providing the names of the parametric models which should be fitted to the data. Default: c("logistic", "richards", "gompertz", "gompertz.exp", "huang", "baranyi").
dr.method	(Character) Define the method used to perform a dose-response analysis: smooth spline fit ("spline") or model fitting ("model").
dr.model	(Character) Provide a list of models from the R package 'drc' to include in the dose-response analysis (if dr.method = "model"). If more than one model is provided, the best-fitting model will be chosen based on the Akaike Information Criterion.
growth.thresh	(Numeric) Define a threshold for growth. Only if any density value in a sample is greater than growth.thresh (default: 1.5) times the start density, further computations are performed. Else, a message is returned.
dr.have.atleast	(Numeric) 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.
dr.parameter	(Character or numeric) The response parameter in the output table to be used for creating a dose response curve. See <a href="#">growth.drFit</a> for further details. Default: "mu.linfit", which represents the maximum slope of the linear regression. Typical options include: "mu.linfit", "lambda.linfit", "dY.linfit", "mu.spline", "dY.spline", "mu.model", and "A.model".
smooth.dr	(Numeric) Smoothing parameter used in the spline fit by smooth.spline during dose response curve estimation. Usually (not necessary) in (0; 1]. See documentation of smooth.spline for further details. Default: NULL.
log.x.dr	(Logical) Indicates whether $\ln(x+1)$ should be applied to the concentration data of the dose response curves. Default: FALSE.
log.y.dr	(Logical) Indicates whether $\ln(y+1)$ should be applied to the response data of the dose response curves. Default: FALSE.
nboot.dr	(Numeric) Defines the number of bootstrap samples for EC50 estimation. Use nboot.dr = 0 to disable bootstrapping. Default: 0.
report	(Character or NULL) Create a PDF ('pdf') and/or HTML ('html') report after running all computations. Define NULL if no report should be created. Default: c('pdf', 'html')
out.dir	Character or NULL Define the name of a folder in which all result files are stored. If NULL, the folder will be named with a combination of "GrowthResults_" and the current date and time.
out.nm	Character or NULL Define the name of the report files. If NULL, the files will be named with a combination of "GrowthReport_" and the current date and time.
export.fig	(Logical) Export all figures created in the report as separate PNG and PDF files (TRUE) or not (FALSE).

<code>export.res</code>	(Logical) Create tab-separated TXT files containing calculated growth parameters and dose-response analysis results as well as an .RData file for the resulting grodata object at the end of the workflow.
<code>...</code>	Further arguments passed to the shiny app.

### Value

A grofit object that contains all computation results, compatible with various plotting functions of the QurvE package and with [growth.report](#).

<code>time</code>	Raw time matrix passed to the function as time (if no grofit object is provided).
<code>data</code>	Raw density dataframe passed to the function as data (if no grofit object is provided).
<code>gcFit</code>	gcFit object created with the call of <a href="#">growth.gcFit</a> .
<code>drFit</code>	drFit object created with the call of <a href="#">growth.drFit</a> .
<code>expdesign</code>	Experimental design table inherited from grodata or created from the identifier columns (columns 1-3) in data.
<code>control</code>	Object of class grofit.control created with the call of <a href="#">growth.control</a> .

### See Also

Other workflows: [flFit\(\)](#), [growth.gcFit\(\)](#)

Other growth fitting functions: [growth.drFit\(\)](#), [growth.gcBootSpline\(\)](#), [growth.gcFitLinear\(\)](#), [growth.gcFitModel\(\)](#), [growth.gcFitSpline\(\)](#), [growth.gcFit\(\)](#)

Other dose-response analysis functions: [flFit\(\)](#), [growth.drBootSpline\(\)](#), [growth.drFitSpline\(\)](#), [growth.gcFit\(\)](#)

---

<code>match_arg</code>	<i>match arg to choices</i>
------------------------	-----------------------------

---

### Description

Wrapper around [match.arg](#) that defaults to ignoring case and trimming white space

### Usage

```
match_arg(arg, choices, multiple = FALSE, ignore_case = TRUE, trim_ws = FALSE)
```

### Arguments

<code>arg</code>	a character vector (of length one unless several.ok is TRUE) or NULL
<code>choices</code>	a character vector of candidate values
<code>multiple</code>	logical specifying if arg should be allowed to have more than one element. Defaults to FALSE
<code>ignore_case</code>	logical indicating whether to ignore capitalization. Defaults to TRUE
<code>trim_ws</code>	logical indicating whether to trim surrounding white space. Defaults to TRUE

### Value

Value(s) matched via partial matching.

---

numberInput	Create a numeric input
-------------	------------------------

---

### Description

Create an input control for entry of numeric values. This is identical to `shiny::numericInput()` but is more flexible in **not** requiring an initial value and in allowing placeholders.

### Usage

```
numberInput(  
  inputId,  
  label,  
  value = NULL,  
  min = NA,  
  max = NA,  
  step = NA,  
  placeholder = NULL,  
  width = NULL  
)
```

### Arguments

inputId	The input slot that will be used to access the value.
label	Display label for the control, or NULL for no label.
value	Initial value. NULL by default.
min	Minimum allowed value
max	Maximum allowed value
step	Interval to use when stepping between min and max
placeholder	A character string giving the user a hint as to what can be entered into the control. Internet Explorer 8 and 9 do not support this option.
width	The width of the input, e.g. '400px', or '100%'; see <code>validateCssUnit()</code> .

### Value

A numeric input control that can be added to a UI definition.

### Server value

A numeric vector of length 1.

### References

Trattner, J. (2021) shinysurveys: Create and Deploy Surveys in 'Shiny' (R package version 0.2.0)

### See Also

`shiny::updateNumericInput()`

## Examples

```
if (interactive()) {
  library(shiny)
  library(shinysurveys)

  ui <- fluidPage(
    numberInput("obs", "Observations:", placeholder = "How many do you see?", min = 1, max = 100),
    verbatimTextOutput("value")
  )
  server <- function(input, output) {
    output$value <- renderText({ input$obs })
  }
  shinyApp(ui, server)
}
```

---

parse_data	<i>Parse raw plate reader data and convert it to a format compatible with QurvE</i>
------------	---

---

## Description

parse\_data takes a raw export file from a plate reader experiment, extracts relevant information and parses it into the format required to run [growth.workflow](#).

## Usage

```
parse_data(
  data.file = NULL,
  map.file = NULL,
  software = c("Gen5", "Gen6", "Biolector", "Chi.Bio", "GrowthProfiler", "Tecan",
    "VictorNivo", "VictorX3"),
  convert.time = NULL,
  sheet.data = 1,
  sheet.map = 1,
  csvsep.data = ";",
  dec.data = ".",
  csvsep.map = ";",
  dec.map = ".",
  subtract.blank = T,
  calibration = NULL
)
```

## Arguments

data.file	(Character) A table file with extension '.xlsx', '.xls', '.csv', '.tsv', or '.txt' containing plate reader data.
map.file	(Character) A table file in column format with 'well', 'ID', 'replicate', and 'concentration' in the first row. Used to assign sample information to wells in a plate.
software	(Character) The name of the software/device used to export the plate reader data.

convert.time	(NULL or string) Convert time values with a formula provided in the form 'y = function(x)'. For example: convert.time = 'y = 24 * x'
sheet.data	(Numeric or Character) Number or name of a sheet in XLS or XLSX files containing experimental data ( <i>optional</i> ).
sheet.map	(Numeric or Character) Number or name of a sheet in XLS or XLSX files containing experimental data ( <i>optional</i> ).
csvsep.data	(Character) separator used in CSV data file (ignored for other file types). Default: ","
dec.data	(Character) decimal separator used in CSV, TSV or TXT data file.
csvsep.map	(Character) separator used in CSV mapping file (ignored for other file types). Default: ","
dec.map	(Character) decimal separator used in CSV, TSV or TXT mapping file.
subtract.blank	(Logical) Shall blank values be subtracted from values within the same experiment ( <a href="#">TRUE</a> , the default) or not ( <a href="#">FALSE</a> ).
calibration	(Character or NULL) Provide an equation in the form 'y = function(x)' (for example: 'y = x^2 * 0.3 - 0.5') to convert density and fluorescence values. This can be used to, e.g., convert plate reader absorbance values into $OD_{600}$ . Caution!: When utilizing calibration, carefully consider whether or not blanks were subtracted to determine the calibration before selecting the input subtract.blank = TRUE.

### Value

A grodata object suitable to run [growth.workflow](#). See [read\\_data](#) for object structure.

---

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

---

### Description

Generic plot function for gcBootSpline objects.

### Usage

```
## S3 method for class 'drBootSpline'
plot(
  x,
  pch = 19,
  colData = 1,
  colSpline = scales::alpha("black", 0.15),
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  lwd = 2,
  plot = TRUE,
  export = FALSE,
  height = 7,
  width = 9,
  out.dir = NULL,
```

```

    combine = FALSE,
    ...
)

```

### Arguments

x	A drBootSpline object created with <a href="#">growth.drBootSpline</a> or stored within a grofit or drFit object created with <a href="#">growth.workflow</a> or <a href="#">growth.drFit</a> , respectively.
pch	(Numeric) Shape of the raw data symbols.
colData	(Numeric or Character) Color used to plot the raw data.
colSpline	(Numeric or Character) Color used to plot the splines.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
lwd	(Numeric) Spline line width.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
combine	(Logical) Indicate whether both dose-response curves and parameter plots shall be shown within the same window.
...	Further arguments to refine the generated base R plot.

---

plot.drFit

*Generic plot function for drFit objects.*


---

### Description

codeplot.drFit calls codeplot.drFitSpline for each group used in a dose-response analysis

### Usage

```

## S3 method for class 'drFit'
plot(
  x,
  combine = TRUE,
  names = NULL,
  exclude.nm = NULL,
  pch = 16,
  cex.point = 2,
  basesize = 15,
  colors = NULL,
  lwd = 0.7,

```

```

    ec50line = TRUE,
    y.lim = NULL,
    x.lim = NULL,
    y.title = NULL,
    x.title = NULL,
    log.y = FALSE,
    log.x = FALSE,
    plot = TRUE,
    export = FALSE,
    height = NULL,
    width = NULL,
    out.dir = NULL,
    out.nm = NULL,
    ...
)

```

### Arguments

x	object of class drFit, created with <a href="#">growth.drFit</a> .
combine	(Logical) Combine the dose-response analysis results of all conditions into a single plot (TRUE) or not (FALSE).
names	(String or vector of strings) Define conditions to combine into a single plot (if combine = TRUE). Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
exclude.nm	(String or vector of strings) Define conditions to exclude from the plot (if combine = TRUE). Partial matches with sample/group names are accepted.
pch	(Numeric) Shape of the raw data symbols.
cex.point	(Numeric) Size of the raw data points.
basesize	(Numeric) Base font size.
colors	(Numeric or character) Define colors for different conditions.
lwd	(Numeric) Line width of the individual splines.
ec50line	(Logical) Show pointed horizontal and vertical lines at the EC50 values (TRUE) or not (FALSE).
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.title	(Character) Optional: Provide a title for the y-axis.
x.title	(Character) Optional: Provide a title for the x-axis.
log.y	(Logical) Log-transform the y-axis of the plot (TRUE) or not (FALSE)?
log.x	(Logical) Log-transform the x-axis of the plot (TRUE) or not (FALSE)?
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).

height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

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

---

## Description

Generic plot function for drFitFLModel objects.

## Usage

```
## S3 method for class 'drFitFLModel'
plot(
  x,
  ec50line = TRUE,
  log = c("xy"),
  pch = 1,
  colSpline = 1,
  colData = 1,
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  y.lim = NULL,
  x.lim = NULL,
  lwd = 2,
  plot = TRUE,
  export = FALSE,
  height = 7,
  width = 9,
  out.dir = NULL,
  ...
)
```

## Arguments

x	Object of class drFitFLModel, created with <a href="#">fl.drFitModel</a> .
ec50line	(Logical) Show pointed horizontal and vertical lines at the EC50 value (TRUE) or not (FALSE).
log	("x", "y", or "xy") Display the x- or y-axis on a logarithmic scale.
pch	(Numeric) Size of the raw data circles.
colSpline	(Numeric or Character) Color used to plot the splines.



colData	(Numeric or Character) Color used to plot the raw data.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis as a vector in the form c(l, u).
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis as a vector in the form c(l, u).
lwd	(Numeric) Line width.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated base R plot.

---

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

---

## Description

Generic plot function for drFitModel objects.

## Usage

```
## S3 method for class 'drFitModel'
plot(
  x,
  type = c("confidence", "all", "bars", "none", "obs", "average"),
  ec50line = TRUE,
  add = FALSE,
  broken = TRUE,
  bp,
  gridsize = 200,
  log = "x",
  n.xbreaks,
  n.ybreaks,
  x.lim,
  y.lim,
  pch = 1,
  cex.point,
  cex.axis = 1,
  cex.lab = 1.3,
  col = 1,
  lwd = 2,
```

```

lty = 2,
xlab,
ylab,
legend = T,
legendText,
legendPos,
cex.legend = NULL,
...
)

```

### Arguments

x	object of class <code>drFitModel</code> , created with <code>growth.drFitModel</code> .
type	(Character) Specify how to plot the data. There are currently 5 options: "average" (averages and fitted curve(s); default), "none" (only the fitted curve(s)), "obs" (only the data points), "all" (all data points and fitted curve(s)), "bars" (averages and fitted curve(s) with model-based standard errors (see Details)), and "confidence" (confidence bands for fitted curve(s)).
ec50line	(Logical) Show pointed horizontal and vertical lines at the EC50 values (TRUE) or not (FALSE).
add	(Logical) If TRUE then add to already existing plot.
broken	(Logical) If TRUE the x axis is broken provided this axis is logarithmic (using functionality in the CRAN package 'plotrix').
bp	(Numeric) Specifying the break point below which the dose is zero (the amount of stretching on the dose axis above zero in order to create the visual illusion of a logarithmic scale including 0). The default is the base-10 value corresponding to the rounded value of the minimum of the log10 values of all positive dose values. This argument is only working for logarithmic dose axes.
gridsize	(Numeric) Number of points in the grid used for plotting the fitted curves.
log	(Character) String which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic. The default is "x". The empty string "" yields the original axes.
n.xbreaks	(Numeric) Number of breaks on the x-axis (if not log-transformed). The breaks are generated using <code>pretty</code> . Thus, the final number of breaks can deviate from the user input.
n.ybreaks	(Numeric) Number of breaks on the y-axis (if not log-transformed). The breaks are generated using <code>pretty</code> . Thus, the final number of breaks can deviate from the user input.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis of both growth curve and derivative plots as a vector in the form <code>c(l, u)</code> . If only the lower or upper bound should be fixed, provide <code>c(l, NA)</code> or <code>c(NA, u)</code> , respectively.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis of the growth curve plot as a vector in the form <code>c(l, u)</code> . If only the lower or upper bound should be fixed, provide <code>c(l, NA)</code> or <code>c(NA, u)</code> , respectively.
pch	(Numeric) Symbol used to plot data points.
cex.point	(Numeric) Size of the raw data points.
cex.axis	(Numeric) Font size of axis annotations.

cex.lab	(Numeric) Font size of axis titles.
col	(Logical or a vector of colors) If TRUE default colours are used. If FALSE (default) no colors are used.
lwd	(Numeric) Line width.
lty	(Numeric) Specify the line type.
xlab	(Character) An optional label for the x axis.
ylab	(Character) An optional label for the y axis.
legend	(Logical) If TRUE a legend is displayed.
legendText	(Character) Specify the legend text (the position of the upper right corner of the legend box).
legendPos	(Numeric) Vector of length 2 giving the position of the legend.
cex.legend	numeric specifying the legend text size.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

## References

Christian Ritz, Florent Baty, Jens C. Streibig, Daniel Gerhard (2015). *Dose-Response Analysis Using R*. PLoS ONE 10(12): e0146021. DOI: 10.1371/journal.pone.0146021

---

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

---

## Description

codeplot.drFitSpline generates the spline fit plot for response-parameter vs. concentration data

## Usage

```
## S3 method for class 'drFitSpline'
plot(
  x,
  add = FALSE,
  ec50line = TRUE,
  log = "",
  pch = 16,
  colSpline = 1,
  colData = 1,
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  y.lim = NULL,
  x.lim = NULL,
  y.title = NULL,
  x.title = NULL,
  lwd = 2,
  plot = TRUE,
  export = FALSE,
```

```

    height = 7,
    width = 9,
    out.dir = NULL,
    ...
)

```

## Arguments

x	object of class drFitSpline, created with <a href="#">growth.drFitSpline</a> .
add	(Logical) Shall the fitted spline be added to an existing plot? TRUE is used internally by <a href="#">plot.drBootSpline</a> .
ec50line	(Logical) Show pointed horizontal and vertical lines at the EC50 value (TRUE) or not (FALSE).
log	("x", "y", or "xy") Display the x- or y-axis on a logarithmic scale.
pch	(Numeric) Shape of the raw data symbols.
colSpline	(Numeric or character) Spline line colour.
colData	(Numeric or character) Contour color of the raw data circles.
cex.point	(Numeric) Size of the raw data symbols.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.title	(Character) Optional: Provide a title for the y-axis.
x.title	(Character) Optional: Provide a title for the x-axis.
lwd	(Numeric) Line width of spline.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated base R plot.

---

plot.dr_parameter	<i>Compare calculated dose-response parameters between conditions.</i>
-------------------	--

---

## Description

plot.dr\_parameter gathers parameters from the results of a dose-response analysis and compares a chosen parameter between each condition in a column plot. Error bars represent the 95% confidence interval (only shown for > 2 replicates).

## Usage

```
## S3 method for class 'dr_parameter'
plot(
  x,
  param = c("y.max", "y.min", "fc", "K", "n", "EC50", "yEC50", "drboot.meanEC50",
    "drboot.meanEC50y"),
  names = NULL,
  exclude.nm = NULL,
  basesize = 12,
  reference.nm = NULL,
  label.size = NULL,
  plot = T,
  export = F,
  height = 7,
  width = NULL,
  out.dir = NULL,
  out.nm = NULL,
  ...
)
```

## Arguments

x	A grofit, drFit, drTable, or flFitRes object obtained with <a href="#">growth.workflow</a> , <a href="#">growth.drFit</a> , <a href="#">fl.drFit</a> , or <a href="#">fl.workflow</a> .
param	(Character) The parameter used to compare different sample groups. Any name of a column containing numeric values in gcTable (which is stored within grofit or gcFit objects) can be used as input. Useful options are: 'y.max', 'y.min', 'fc', 'K', or 'n' for fluorescence dose-response analyses with dr.type = 'model' in the control argument, or 'EC50', 'yEC50', 'drboot.meanEC50', 'drboot.meanEC50y'.
names	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
exclude.nm	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
basesize	(Numeric) Base font size.

reference.nm	(Character) Name of the reference condition, to which parameter values are normalized. Partially matching strings are tolerated as long as they can uniquely identify the condition.
label.size	(Numeric) Font size for sample labels below x-axis.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

plot.dual

---

*Compare fluorescence and density over time*


---

## Description

plot.dual creates a two-panel plot in which fluorescence or density values are shown over time, allowing for the identification of, e.g., expression patterns in different growth stages.

## Usage

```
## S3 method for class 'dual'
plot(
  x,
  fluorescence = c("fl", "norm.fl"),
  IDs = NULL,
  names = NULL,
  conc = NULL,
  mean = TRUE,
  exclude.nm = NULL,
  exclude.conc = NULL,
  log.y.density = F,
  log.y.fl = F,
  n.ybreaks = 6,
  colors = NULL,
  basesize = 20,
  y.lim.density = NULL,
  y.lim.fl = NULL,
  x.lim = NULL,
  x.title = NULL,
  y.title.density = NULL,
  y.title.fl = NULL,
```

```

    lwd = 1.1,
    legend.position = "bottom",
    legend.ncol = 2,
    plot = TRUE,
    export = FALSE,
    height = NULL,
    width = NULL,
    out.dir = NULL,
    out.nm = NULL,
    shiny = FALSE,
    ...
)

```

## Arguments

x	A flFit, flFitRes, or grodata object created with <a href="#">flFit</a> , <a href="#">fl.workflow</a> or <a href="#">read_data</a>
fluorescence	(Character) Indicate, which type of fluorescence data should be displayed.
IDs	(String or vector of strings) Define samples or groups (if mean = TRUE) to combine into a single plot based on exact matches with entries in the label or condition columns of grofit\$expdesign.
names	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
conc	(Numeric or numeric vector) Define concentrations to combine into a single plot. If NULL, all concentrations are considered. Note: Ensure to use unique concentration values to extract groups of interest. If the concentration value of one condition is included in its entirety within the name of other conditions (e.g., the dataset contains '1', '10', and '100', code = 10 will select both '10 and '100'), it cannot be extracted individually.
mean	(Logical) Display the mean and standard deviation of groups with replicates (TRUE) or plot each sample individually (FALSE)?
exclude.nm	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
exclude.conc	(Numeric or numeric vector) Define concentrations to exclude from the plot.
log.y.density	(Logical) Log-transform the y-axis of the density plot (TRUE) or not (FALSE)?
log.y.fl	(Logical) Log-transform the y-axis of the fluorescence plot (TRUE) or not (FALSE)?
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using <code>scales::pretty_breaks</code> . Thus, the final number of breaks can deviate from the user input.
colors	(vector of strings) Define a color palette used to draw the plots. If NULL, default palettes are chosen based on the number of groups/samples within the plot. Note: The number of provided colors should at least match the number of groups/samples.
basesize	(Numeric) Base font size.

y.lim.density	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the y-axis of the density plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.lim.fl	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the y-axis of the fluorescence plot as a vector in the form c(l, u).
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the x-axis of both fluorescence and density plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.title	(Character) Optional: Provide a title for the x-axis of both growth curve and derivative plots.
y.title.density	(Character) Optional: Provide a title for the y-axis of the density plot.
y.title.fl	(Character) Optional: Provide a title for the y-axis of the fluorescence plot.
lwd	(Numeric) Line width of the individual plots.
legend.position	(Character) Position of the legend. One of "bottom", "top", "left", "right".
legend.ncol	(Numeric) Number of columns in the legend.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
shiny	(Logical) Indicate if plot is generated within the shiny app.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

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

---

## Description

Generic plot function for flBootSpline objects.



**Usage**

```
## S3 method for class 'flBootSpline'
plot(
  x,
  pch = 1,
  colData = 1,
  deriv = TRUE,
  colSpline = ggplot2::alpha("dodgerblue3", 0.2),
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  lwd = 2,
  y.lim = NULL,
  x.lim = NULL,
  y.lim.deriv = NULL,
  plot = TRUE,
  export = FALSE,
  height = 7,
  width = 9,
  out.dir = NULL,
  combine = FALSE,
  ...
)
```

**Arguments**

x	Object of class flBootSpline, created with <a href="#">flBootSpline</a> .
pch	(Numeric) Size of the raw data circles.
colData	(Numeric or Character) Color used to plot the raw data.
deriv	(Logical) Show the derivatives (i.e., slope) over time in a secondary plot (TRUE) or not (FALSE).
colSpline	(Numeric or Character) Color used to plot the splines.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
lwd	(Numeric) Spline line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis of the fluorescence curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis of both fluorescence curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).

export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
combine	(Logical) Indicate whether both growth curves and parameter plots shall be shown within the same window.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

plot.flFitLinear	<i>Generic plot function for flcFittedLinear objects. Plot the results of a linear regression on ln-transformed data</i>
------------------	--

---

## Description

plot.flFitLinear shows the results of a linear regression and visualizes raw data, data points included in the fit, the tangent obtained by linear regression, and the lag time.

## Usage

```
## S3 method for class 'flFitLinear'
plot(
  x,
  log = "",
  which = c("fit", "diagnostics", "fit_diagnostics"),
  pch = 21,
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  lwd = 2,
  y.lim = NULL,
  x.lim = NULL,
  plot = TRUE,
  export = FALSE,
  height = ifelse(which == "fit", 7, 5),
  width = ifelse(which == "fit", 9, 9),
  out.dir = NULL,
  ...
)
```

## Arguments

x	A flFittedLinear object created with <a href="#">flFitLinear</a> or stored within a flFitRes or flFit object created with <a href="#">fl.workflow</a> or <a href="#">flFit</a> , respectively.
log	("x" or "y") Display the x- or y-axis on a logarithmic scale.
which	("fit" or "diagnostics") Display either the results of the linear fit on the raw data or statistical evaluation of the linear regression.

pch	(Numeric) Shape of the raw data symbols.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
lwd	(Numeric) Line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis as a vector in the form c(l, u).
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis as a vector in the form c(l, u).
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated base R plot.

---

plot.flFitRes

---

Combine different groups of samples into a single plot

---

## Description

Visualize fluorescence, normalized fluorescence, or spline fits of multiple sample groups in a single plot.

## Usage

```
## S3 method for class 'flFitRes'
plot(
  x,
  data.type = c("spline", "raw", "norm.fl"),
  IDs = NULL,
  names = NULL,
  conc = NULL,
  mean = TRUE,
  exclude.nm = NULL,
  exclude.conc = NULL,
  log.y = F,
  deriv = F,
  n.ybreaks = 6,
  colors = NULL,
  basesize = 20,
  y.lim = NULL,
  x.lim = NULL,
  y.title = NULL,
  x.title = NULL,
```

```

    y.lim.deriv = NULL,
    y.title.deriv = NULL,
    lwd = 1.1,
    legend.position = "bottom",
    legend.ncol = 2,
    plot = TRUE,
    export = FALSE,
    height = NULL,
    width = NULL,
    out.dir = NULL,
    out.nm = NULL,
    shiny = FALSE,
    ...
)

## S3 method for class 'flFit'
plot(
  x,
  data.type = c("spline", "raw", "norm.fl"),
  IDs = NULL,
  names = NULL,
  conc = NULL,
  mean = TRUE,
  exclude.nm = NULL,
  exclude.conc = NULL,
  log.y = F,
  deriv = F,
  n.ybreaks = 6,
  colors = NULL,
  basesize = 20,
  y.lim = NULL,
  x.lim = NULL,
  y.title = NULL,
  x.title = NULL,
  y.lim.deriv = NULL,
  y.title.deriv = NULL,
  lwd = 1.1,
  legend.position = "bottom",
  legend.ncol = 2,
  plot = TRUE,
  export = FALSE,
  height = NULL,
  width = NULL,
  out.dir = NULL,
  out.nm = NULL,
  shiny = FALSE,
  ...
)

```

### Arguments

**x** A `flFitRes`, `flFit`, or `grodata` object created with [fl.workflow](#) containing fluorescence data.

data.type	(Character) Indicate, which type of fluorescence data should be displayed.
IDs	(String or vector of strings) Define samples or groups (if mean = TRUE) to combine into a single plot based on exact matches with entries in the label or condition columns of grofit\$expdesign.
names	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
conc	(Numeric or numeric vector) Define concentrations to combine into a single plot. If NULL, all concentrations are considered. Note: Ensure to use unique concentration values to extract groups of interest. If the concentration value of one condition is included in its entirety within the name of other conditions (e.g., the dataset contains '1', '10', and '100', code = 10 will select both '10 and '100'), it cannot be extracted individually.
mean	(Logical) Display the mean and standard deviation of groups with replicates (TRUE) or plot each sample individually (FALSE)?
exclude.nm	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
exclude.conc	(Numeric or numeric vector) Define concentrations to exclude from the plot.
log.y	(Logical) Log-transform the y-axis of the plot (TRUE) or not (FALSE)?
deriv	(Logical) Show derivatives over time in a separate panel below the plot (TRUE) or not (FALSE)?
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using axisTicks(). Thus, the final number of breaks can deviate from the user input.
colors	(vector of strings) Define a color palette used to draw the plots. If NULL, default palettes are chosen based on the number of groups/samples within the plot. Note: The number of provided colors should at least match the number of groups/samples.
basesize	(Numeric) Base font size.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the y-axis of the fluorescence curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the x-axis of both fluorescence curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.title	(Character) Optional: Provide a title for the y-axis of the fluorescence curve plot.
x.title	(Character) Optional: Provide a title for the x-axis of both fluorescence curve and derivative plots.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.title.deriv	(Character) Optional: Provide a title for the y-axis of the derivative plot.

lwd	(Numeric) Line width of the individual plots.
legend.position	(Character) Position of the legend. One of "bottom", "top", "left", "right".
legend.ncol	(Numeric) Number of columns in the legend.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
shiny	(Logical) Indicate if plot is generated within the shiny app.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

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

---

## Description

plot.flFitSpline generates the spline fit plot for a single sample.

## Usage

```
## S3 method for class 'flFitSpline'
plot(
  x,
  add = FALSE,
  raw = TRUE,
  slope = TRUE,
  deriv = T,
  spline = T,
  log.y = F,
  basesize = 16,
  pch = 1,
  colData = 1,
  colSpline = "dodgerblue3",
  cex.point = 2,
  lwd = 0.7,
  y.lim = NULL,
  x.lim = NULL,
  y.lim.deriv = NULL,
  n.ybreaks = 6,
  plot = TRUE,
  export = FALSE,
```

```

width = 8,
height = ifelse(deriv == TRUE, 8, 6),
out.dir = NULL,
...
)

```

## Arguments

x	Object of class flFitSpline, created with <a href="#">flFitSpline</a> .
add	(Logical) Shall the fitted spline be added to an existing plot? TRUE is used internally by <a href="#">plot.flBootSpline</a> .
raw	(Logical) Display raw density as circles (TRUE) or not (FALSE).
slope	(Logical) Show the slope at the maximum slope (TRUE) or not (FALSE).
deriv	(Logical) Show the derivative (i.e., slope) over time in a secondary plot (TRUE) or not (FALSE).
spline	(Logical) Only for add = TRUE: add the current spline to the existing plot (FALSE).
log.y	(Logical) Log-transform the y-axis (TRUE) or not (FALSE).
basesize	(Numeric) Base font size.
pch	(Numeric) Size of the raw data circles.
colData	(Numeric or character) Contour color of the raw data circles.
colSpline	(Numeric or character) Spline line colour.
cex.point	(Numeric) Size of the raw data points.
lwd	(Numeric) Spline line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis of the fluorescence curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis of both fluorescence curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using <code>axisTicks()</code> . Thus, the final number of breaks can deviate from the user input.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
width	(Numeric) Width of the exported image in inches.
height	(Numeric) Height of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

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

---

## Description

Generic plot function for gcBootSpline objects.

## Usage

```
## S3 method for class 'gcBootSpline'
plot(
  x,
  pch = 1,
  colData = 1,
  deriv = TRUE,
  colSpline = ggplot2::alpha("dodgerblue3", 0.2),
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  lwd = 2,
  y.lim = NULL,
  x.lim = NULL,
  y.lim.deriv = NULL,
  plot = TRUE,
  export = FALSE,
  height = 7,
  width = 9,
  out.dir = NULL,
  combine = FALSE,
  ...
)
```

## Arguments

x	object of class gcBootSpline, created with <a href="#">growth.gcBootSpline</a> .
pch	(Numeric) Symbol used to plot data points.
colData	(Numeric or character) Contour color of the raw data circles.
deriv	(Logical) Show the derivatives (i.e., slope) over time in a secondary plot (TRUE) or not (FALSE).
colSpline	(Numeric or character) Spline line colour.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
lwd	(Numeric) Spline line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis of the growth curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.



x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis of both growth curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
combine	(Logical) Indicate whether both growth curves and parameter plots shall be shown within the same window.
...	Further arguments to refine the generated base R plot.

---

plot.gcFitLinear	<i>Generic plot function for gcFittedLinear objects. Plot the results of a linear regression on ln-transformed data</i>
------------------	---

---

## Description

plot.gcFitLinear shows the results of a linear regression on log-transformed data and visualizes raw data, data points included in the fit, the tangent obtained by linear regression, and the lag time.

## Usage

```
## S3 method for class 'gcFitLinear'
plot(
  x,
  log = "y",
  which = c("fit", "diagnostics", "fit_diagnostics"),
  pch = 21,
  cex.point = 1,
  cex.lab = 1.5,
  cex.axis = 1.3,
  lwd = 2,
  y.lim = NULL,
  x.lim = NULL,
  plot = TRUE,
  export = FALSE,
  height = ifelse(which == "fit", 7, 5),
  width = ifelse(which == "fit", 9, 9),
  out.dir = NULL,
  ...
)
```

**Arguments**

x	A gcFittedLinear object created with <code>growth.gcFitLinear</code> or stored within a grofit or gcFit object created with <code>growth.workflow</code> or <code>growth.gcFit</code> , respectively.
log	("x" or "y") Display the x- or y-axis on a logarithmic scale.
which	("fit" or "diagnostics") Display either the results of the linear fit on the raw data or statistical evaluation of the linear regression.
pch	(Numeric) Shape of the raw data symbols.
cex.point	(Numeric) Size of the raw data points.
cex.lab	(Numeric) Font size of axis titles.
cex.axis	(Numeric) Font size of axis annotations.
lwd	(Numeric) Line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis as a vector in the form <code>c(l, u)</code> .
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis as a vector in the form <code>c(l, u)</code> .
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE).
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated base R plot.

---

plot.gcFitModel

*Generic plot function for gcBootSpline objects.*


---

**Description**

Plot the results of a parametric model fit on density vs. time data

**Usage**

```
## S3 method for class 'gcFitModel'
plot(
  x,
  raw = TRUE,
  pch = 1,
  colData = 1,
  equation = TRUE,
  eq.size = 1,
  colModel = ggplot2::alpha("forestgreen", 0.85),
  basesize = 16,
  cex.point = 2,
  lwd = 0.7,
```

```

    n.ybreaks = 6,
    plot = TRUE,
    export = FALSE,
    height = 8,
    width = 6,
    out.dir = NULL,
    ...
)

```

### Arguments

x	A gcFittedModel object created with <a href="#">growth.gcFitModel</a> or stored within a grofit or gcFit object created with <a href="#">growth.workflow</a> or <a href="#">growth.gcFit</a> , respectively.
raw	(Logical) Show the raw data within the plot (TRUE) or not (FALSE).
pch	(Numeric) Symbol used to plot data points.
colData	(Numeric or Character) Color used to plot the raw data.
equation	(Logical) Show the equation of the fitted model within the plot (TRUE) or not (FALSE).
eq.size	(Numeric) Provide a value to scale the size of the displayed equation.
colModel	(Numeric or Character) Color used to plot the fitted model.
basesize	(Numeric) Base font size.
cex.point	(Numeric) Size of the raw data points.
lwd	(Numeric) Spline line width.
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using <code>scales::pretty_breaks</code> . Thus, the final number of breaks can deviate from the user input.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated ggplot2 plot.

---

plot.gcFitSpline

*Generic plot function for gcFitSpline objects.*


---

### Description

codeplot.gcFitSpline generates the spline fit plot for a single sample.

**Usage**

```
## S3 method for class 'gcFitSpline'
plot(
  x,
  add = FALSE,
  raw = TRUE,
  slope = TRUE,
  deriv = T,
  spline = T,
  log.y = T,
  pch = 1,
  colData = 1,
  colSpline = "dodgerblue3",
  basesize = 16,
  cex.point = 2,
  lwd = 0.7,
  y.lim = NULL,
  x.lim = NULL,
  y.lim.deriv = NULL,
  n.ybreaks = 6,
  plot = TRUE,
  export = FALSE,
  width = 8,
  height = ifelse(deriv == TRUE, 8, 6),
  out.dir = NULL,
  ...
)
```

**Arguments**

x	object of class gcFitSpline, created with <a href="#">growth.gcFitSpline</a> .
add	(Logical) Shall the fitted spline be added to an existing plot? TRUE is used internally by <a href="#">plot.gcBootSpline</a> .
raw	(Logical) Display raw density as circles (TRUE) or not (FALSE).
slope	(Logical) Show the slope at the maximum growth rate (TRUE) or not (FALSE).
deriv	(Logical) Show the derivative (i.e., slope) over time in a secondary plot (TRUE) or not (FALSE).
spline	(Logical) Only for add = TRUE: add the current spline to the existing plot (FALSE).
log.y	(Logical) Log-transform the y-axis (TRUE) or not (FALSE).
pch	(Numeric) Symbol used to plot data points.
colData	(Numeric or character) Contour color of the raw data circles.
colSpline	(Numeric or character) Spline line colour.
basesize	(Numeric) Base font size.
cex.point	(Numeric) Size of the raw data points.
lwd	(Numeric) Spline line width.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on y-axis of the growth curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.

x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the x-axis of both growth curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using scales::pretty_breaks. Thus, the final number of breaks can deviate from the user input.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
width	(Numeric) Width of the exported image in inches.
height	(Numeric) Height of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
...	Further arguments to refine the generated base R plot (if add = TRUE).

---

plot.grodata	<i>Generic plot function for grodata objects. Plots raw density, fluorescence, or normalized fluorescence data of multiple samples or conditions.</i>
--------------	---

---

## Description

plot.grodata calls link{plot.grofit} or link{plot.flFitRes} based on the chosen data.type, respectively.

## Usage

```
## S3 method for class 'grodata'
plot(
  x,
  data.type = c("dens", "fl", "norm.fl"),
  IDs = NULL,
  names = NULL,
  conc = NULL,
  mean = TRUE,
  exclude.nm = NULL,
  exclude.conc = NULL,
  log.y = F,
  n.ybreaks = 6,
  colors = NULL,
  basesize = 20,
  y.lim = NULL,
  x.lim = NULL,
```

```

y.title = NULL,
x.title = NULL,
lwd = 1.1,
legend.position = "bottom",
legend.ncol = 2,
plot = TRUE,
export = FALSE,
height = NULL,
width = NULL,
out.dir = NULL,
out.nm = NULL,
shiny = FALSE,
...
)

```

### Arguments

x	A grodata object created with <a href="#">read_data</a> or <a href="#">parse_data</a> .
data.type	(Character) Plot either raw density (data.type = "dens"), raw fluorescence (data.type = "fl"), or fluorescence normalized to density (data.type = "norm.fl").
IDs	(String or vector of strings) Define samples or groups (if mean = TRUE) to combine into a single plot based on exact matches with entries in the label or condition columns of grofit\$expdesign.
names	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
conc	(Numeric or numeric vector) Define concentrations to combine into a single plot. If NULL, all concentrations are considered. Note: Ensure to use unique concentration values to extract groups of interest. If the concentration value of one condition is included in its entirety within the name of other conditions (e.g., the dataset contains '1', '10', and '100', code = 10 will select both '10 and '100'), it cannot be extracted individually.
mean	(Logical) Display the mean and standard deviation of groups with replicates (TRUE) or plot each sample individually (FALSE)?
exclude.nm	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
exclude.conc	(Numeric or numeric vector) Define concentrations to exclude from the plot.
log.y	(Logical) Log-transform the y-axis of the plot (TRUE) or not (FALSE)?
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using scales::pretty_breaks. Thus, the final number of breaks can deviate from the user input.
colors	(vector of strings) Define a color palette used to draw the plots. If NULL, default palettes are chosen based on the number of groups/samples within the plot. Note: The number of provided colors should at least match the number of groups/samples.
basesize	(Numeric) Base font size.

y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the y-axis of the growth curve plot as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the x-axis of both growth curve and derivative plots as a vector in the form c(l, u). If only the lower or upper bound should be fixed, provide c(l, NA) or c(NA, u), respectively.
y.title	(Character) Optional: Provide a title for the y-axis of the growth curve plot.
x.title	(Character) Optional: Provide a title for the x-axis of both growth curve and derivative plots.
lwd	(Numeric) Line width of the individual plots.
legend.position	(Character) Position of the legend. One of "bottom", "top", "left", "right".
legend.ncol	(Numeric) Number of columns in the legend.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
shiny	(Logical) Indicate if plot is generated within the shiny app.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

---

plot.grofit	<i>Generic plot function for grofit objects. Combine different groups of samples into a single plot</i>
-------------	---

---

## Description

plot.grofit extracts the spline fits of a subset of samples in a grofit object calculates averages and standard deviations of conditions with replicates and combines them into a single plot.

## Usage

```
## S3 method for class 'grofit'
plot(
  x,
  ...,
  data.type = c("spline", "raw"),
  IDs = NULL,
  names = NULL,
```

```

conc = NULL,
exclude.nm = NULL,
exclude.conc = NULL,
mean = TRUE,
log.y = T,
deriv = T,
n.ybreaks = 6,
colors = NULL,
basesize = 20,
y.lim = NULL,
x.lim = NULL,
y.title = NULL,
x.title = NULL,
y.lim.deriv = NULL,
y.title.deriv = NULL,
lwd = 1.1,
legend.position = "bottom",
legend.ncol = 2,
plot = TRUE,
export = FALSE,
height = NULL,
width = NULL,
out.dir = NULL,
out.nm = NULL,
shiny = FALSE
)

```

### Arguments

<code>x</code>	A grofit object created with <a href="#">growth.workflow</a> containing spline fits.
<code>...</code>	( <i>optional</i> ) Additional grofit objects created in separate workflows.
<code>data.type</code>	(Character) Plot either raw data ( <code>data.type = "raw"</code> ) or the spline fit results
<code>IDs</code>	(String or vector of strings) Define samples or groups (if <code>mean = TRUE</code> ) to combine into a single plot based on exact matches with entries in the label or condition columns of <code>grofit\$expdesign</code> .
<code>names</code>	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If <code>NULL</code> , all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
<code>conc</code>	(Numeric or numeric vector) Define concentrations to combine into a single plot. If <code>NULL</code> , all concentrations are considered. Note: Ensure to use unique concentration values to extract groups of interest. If the concentration value of one condition is included in its entirety within the name of other conditions (e.g., the dataset contains '1', '10', and '100', <code>code = 10</code> will select both '10 and '100'), it cannot be extracted individually.
<code>exclude.nm</code>	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
<code>exclude.conc</code>	(Numeric or numeric vector) Define concentrations to exclude from the plot.
<code>mean</code>	(Logical) Display the mean and standard deviation of groups with replicates ( <code>TRUE</code> ) or plot each sample individually ( <code>FALSE</code> )?



log.y	(Logical) Log-transform the y-axis of the plot (TRUE) or not (FALSE)?
deriv	(Logical) Show derivatives over time in a separate panel below the plot (TRUE) or not (FALSE)?
n.ybreaks	(Numeric) Number of breaks on the y-axis. The breaks are generated using <code>scales::pretty_breaks</code> . Thus, the final number of breaks can deviate from the user input.
colors	(vector of strings) Define a color palette used to draw the plots. If NULL, default palettes are chosen based on the number of groups/samples within the plot. Note: The number of provided colors should at least match the number of groups/samples.
basesize	(Numeric) Base font size.
y.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the y-axis of the growth curve plot as a vector in the form <code>c(l, u)</code> . If only the lower or upper bound should be fixed, provide <code>c(1, NA)</code> or <code>c(NA, u)</code> , respectively.
x.lim	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds of the x-axis of both growth curve and derivative plots as a vector in the form <code>c(l, u)</code> . If only the lower or upper bound should be fixed, provide <code>c(1, NA)</code> or <code>c(NA, u)</code> , respectively.
y.title	(Character) Optional: Provide a title for the y-axis of the growth curve plot.
x.title	(Character) Optional: Provide a title for the x-axis of both growth curve and derivative plots.
y.lim.deriv	(Numeric vector with two elements) Optional: Provide the lower (l) and upper (u) bounds on the y-axis of the derivative plot as a vector in the form <code>c(l, u)</code> . If only the lower or upper bound should be fixed, provide <code>c(1, NA)</code> or <code>c(NA, u)</code> , respectively.
y.title.deriv	(Character) Optional: Provide a title for the y-axis of the derivative plot.
lwd	(Numeric) Line width of the individual plots.
legend.position	(Character) Position of the legend. One of "bottom", "top", "left", "right".
legend.ncol	(Numeric) Number of columns in the legend.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if <code>export = TRUE</code> . If NULL, a name will be automatically generated including the chosen parameter.
shiny	(Logical) Indicate if plot is generated within the shiny app.

plot.parameter

*Compare growth parameters between samples or conditions***Description**

plot.parameter gathers physiological parameters from the results of a growth fit analysis and compares a chosen parameter between each sample or condition in a column plot. Error bars represent the 95% confidence interval (only shown for > 2 replicates).

**Usage**

```
## S3 method for class 'parameter'
plot(
  x,
  param = c("mu.linfit", "lambda.linfit", "dY.linfit", "A.linfit", "mu2.linfit",
    "lambda2.linfit", "mu.model", "lambda.model", "A.model", "A.orig.model", "dY.model",
    "dY.orig.model", "tD.linfit", "tD2.linfit", "tD.spline", "tD2.spline", "mu.spline",
    "lambda.spline", "A.spline", "dY.spline", "integral.spline", "mu2.spline",
    "lambda2.spline", "mu.bt", "lambda.bt", "A.bt", "integral.bt", "max_slope.linfit",
    "max_slope.spline"),
  IDs = NULL,
  names = NULL,
  conc = NULL,
  exclude.nm = NULL,
  exclude.conc = NULL,
  reference.nm = NULL,
  reference.conc = NULL,
  basesize = 12,
  label.size = NULL,
  shape.size = 2.5,
  legend.position = "right",
  legend.ncol = 1,
  plot = T,
  export = F,
  height = 7,
  width = NULL,
  out.dir = NULL,
  out.nm = NULL,
  ...
)
```

**Arguments**

x	A grofit, gcFit, or gcTable object obtained with <a href="#">growth.workflow</a> or <a href="#">growth.gcFit</a> .
param	(Character) The parameter used to compare different sample groups. Any name of a column containing numeric values in gcTable (which is stored within grofit or gcFit objects) can be used as input. Useful options are: 'mu.linfit', 'lambda.linfit', 'dY.linfit', 'A.linfit', 'mu.model', 'lambda.model', 'A.model', 'mu.spline', 'lambda.spline', 'A.spline', 'dY.spline', 'integral.spline', 'mu.bt', 'lambda.bt', 'A.bt', 'integral.bt'

IDs	(String or vector of strings) Define samples or groups (if mean = TRUE) to combine into a single plot based on exact matches with entries in the label or condition columns of <code>grofit\$expdesign</code> .
names	(String or vector of strings) Define groups to combine into a single plot. Partial matches with sample/group names are accepted. If NULL, all samples are considered. Note: Ensure to use unique substrings to extract groups of interest. If the name of one condition is included in its entirety within the name of other conditions, it cannot be extracted individually.
conc	(Numeric or numeric vector) Define concentrations to combine into a single plot. If NULL, all concentrations are considered. Note: Ensure to use unique concentration values to extract groups of interest. If the concentration value of one condition is included in its entirety within the name of other conditions (e.g., the dataset contains '1', '10', and '100', code = 10 will select both '10 and '100'), it cannot be extracted individually.
exclude.nm	(String or vector of strings) Define groups to exclude from the plot. Partial matches with sample/group names are accepted.
exclude.conc	(Numeric or numeric vector) Define concentrations to exclude from the plot.
reference.nm	(Character) Name of the reference condition, to which parameter values are normalized. Partially matching strings are tolerated as long as they can uniquely identify the condition.
reference.conc	(Numeric) Concentration of the reference condition, to which parameter values are normalized.
basesize	(Numeric) Base font size.
label.size	(Numeric) Font size for sample labels below x-axis.
shape.size	(Numeric) The size of the symbols indicating replicate values. Default: 2.5
legend.position	(Character) Position of the legend. One of "bottom", "top", "left", "right".
legend.ncol	(Numeric) Number of columns in the legend.
plot	(Logical) Show the generated plot in the Plots pane (TRUE) or not (FALSE). If FALSE, a ggplot object is returned.
export	(Logical) Export the generated plot as PDF and PNG files (TRUE) or not (FALSE).
height	(Numeric) Height of the exported image in inches.
width	(Numeric) Width of the exported image in inches.
out.dir	(Character) Name or path to a folder in which the exported files are stored. If NULL, a "Plots" folder is created in the current working directory to store the files in.
out.nm	(Character) The name of the PDF and PNG files if export = TRUE. If NULL, a name will be automatically generated including the chosen parameter.
...	Additional arguments. This has currently no effect and is only meant to fulfill the requirements of a generic function.

read\_data

*Read growth and fluorescence data in table format***Description**

read\_data reads table files or R dataframe objects containing growth and fluorescence data and extracts datasets, sample and group information, performs blank correction, applies data transformation (calibration), and combines technical replicates.

**Usage**

```
read_data(
  data.density,
  data.fl = NA,
  data.format = "col",
  csvsep = ";",
  dec = ".",
  csvsep.fl = ";",
  dec.fl = ".",
  sheet.density = 1,
  sheet.fl = 1,
  subtract.blank = T,
  convert.time = NULL,
  calibration = NULL
)
```

**Arguments**

- |              |   |
|--------------|---|
| data.density | An R dataframe object or a table file with extension '.xlsx', '.xls', '.csv', '.tsv', or '.txt' containing density data. In column format, the first three table rows contain <ol style="list-style-type: none"> <li>1. sample description</li> <li>2. replicate number (<i>optional</i>: followed by a letter to indicate technical replicates)</li> <li>3. concentration value (<i>optional</i>)</li> </ol> |
| data.fl      | (optional) An R dataframe object or a table file with extension '.xlsx', '.xls', '.csv', '.tsv', or '.txt' containing fluorescence data. Table layout must mimic that of data.density.  |
| data.format  | (Character) "col" for samples in columns, or "row" for samples in rows. Default: "col"  |
| csvsep       | (Character) separator used in CSV file storing density data (ignored for other file types). Default: ";"  |
| dec          | (Character) decimal separator used in CSV, TSV or TXT file storing density data. Default: "."   |
| csvsep.fl    | (Character) separator used in CSV file storing fluorescence data (ignored for other file types). Default: ";"   |
| dec.fl       | (Character) decimal separator used in CSV, TSV or TXT file storing fluorescence data. Default: "."  |

- `sheet.density` (Numeric or Character) Number or name of the sheet with density data in XLS or XLSX files (*optional*).
- `sheet.fl` (Numeric or Character) Number or name of the sheet with fluorescence data in XLS or XLSX files (*optional*).
- `subtract.blank` (Logical) Shall blank values be subtracted from values within the same experiment (**TRUE**, the default) or not (**FALSE**). If `calibration = TRUE`, blanks are subtracted after value conversion.
- `convert.time` (NULL or string) Convert time values with a formula provided in the form '`y = function(x)`'. For example: `convert.time = 'y = 24 * x'`
- `calibration` (Character or NULL) Provide an equation in the form '`y = function(x)`' (for example: '`y = x^2 * 0.3 - 0.5`') to convert density and fluorescence values. This can be used to, e.g., convert plate reader absorbance values into  $OD_{600}$ . Caution!: When utilizing calibration, carefully consider whether or not blanks were subtracted to determine the calibration before selecting the input `subtract.blank = TRUE`.

## Details

QuvE data layout

different experiments with differing time values are distinguished by the presence of separate 'time' columns

Blank for experiment 1

samples experiment 1

samples experiment 2

Time	Blank	OrigL_1A	OrigL_1B	OrigL_1C	OrigL_1D	OrigL_1E	OrigL_1F	OrigL_1G	OrigL_1H	OrigL_1I	OrigL_1J	Time	OrigL_2A	OrigL_2B	OrigL_2C	OrigL_2D	OrigL_2E	OrigL_2F	OrigL_2G	OrigL_2H	OrigL_2I	OrigL_2J
0.00	0.001	0.097	0.096	0.119	0.122	0.098	0.092	0.097	0.096	0.119	0.122	0.096	0.096	0.096	0.121	0.127	0.094	0.096	0.091	0.098	0.092	0.093
0.17	0.030	0.098	0.095	0.119	0.119	0.099	0.092	0.098	0.095	0.119	0.119	0.096	0.096	0.100	0.1	0.096	0.131	0.131	0.099	0.102	0.1	0.097
0.33	0.033	0.101	0.098	0.12	0.121	0.099	0.092	0.101	0.098	0.12	0.121	0.096	0.096	2.00	0.102	0.097	0.142	0.145	0.108	0.108	0.101	0.102
0.50	0.037	0.105	0.102	0.121	0.121	0.099	0.092	0.105	0.102	0.121	0.121	0.096	0.096	3.00	0.117	0.097	0.201	0.201	0.119	0.117	0.103	0.114
0.67	0.035	0.108	0.105	0.123	0.123	0.099	0.092	0.108	0.105	0.123	0.123	0.096	0.096	4.00	0.303	0.271	0.414	0.401	0.309	0.322	0.331	0.289
0.83	0.033	0.109	0.108	0.125	0.125	0.094	0.093	0.109	0.108	0.125	0.125	0.096	0.096	5.00	0.501	0.48	0.689	0.698	0.515	0.587	0.59	0.511
1.00	0.033	0.111	0.11	0.125	0.125	0.101	0.095	0.111	0.11	0.125	0.125	0.097	0.096	6.00	0.901	0.807	1.199	1.198	0.905	1.048	1.004	0.955
1.17	0.031	0.114	0.111	0.127	0.127	0.105	0.095	0.114	0.111	0.127	0.127	0.098	0.098	7.00	1.582	1.471	2.113	2.282	1.577	1.752	1.774	1.491
1.33	0.034	0.116	0.114	0.128	0.128	0.109	0.097	0.116	0.114	0.128	0.128	0.097	0.096	8.00	2.725	2.475	3.403	3.729	2.789	3.179	2.825	2.663
1.50	0.040	0.12	0.118	0.131	0.131	0.111	0.096	0.12	0.118	0.131	0.131	0.097	0.096	9.00	4.74	4.332	5.918	6.484	5.031	5.51	4.87	4.805

## Value

An R list object of class `grodata` containing a time matrix, dataframes with density and fluorescence data (if applicable), and an experimental design table. The `grodata` object can be directly used to run `growth.workflow/fl.workflow` or, together with a `growth.control/fl.control` object, in `growth.gcfIt/flfIt`.

- `time` Matrix with raw time values extracted from `data.density`.
- `density` Dataframe with raw density values and sample identifiers extracted from `data.density`.
- `fluorescence` Dataframe with raw fluorescence values and sample identifiers extracted from `data.fl`. NA, if no fluorescence data is provided.
- `norm.fluorescence` fluorescence data divided by density values. NA, if no fluorescence data is provided.
- `expdesign` Experimental design table created from the first three identifier rows/columns (see argument `data.format`) (`data.density`).

---

read_file	<i>Call the appropriate function required to read a table file and return the table as a dataframe object.</i>
-----------	--

---

### Description

Call the appropriate function required to read a table file and return the table as a dataframe object.

### Usage

```
read_file(filename, csvsep = ";", dec = ".", sheet = 1)
```

### Arguments

filename	(Character) Name or path of the table file to read. Can be of type CSV, XLS, XLSX, TSV, or TXT.
csvsep	(Character) separator used in CSV file (ignored for other file types).
dec	(Character) decimal separator used in CSV, TSV and TXT files.
sheet	(Numeric or Character) Number or name of a sheet in XLS or XLSX files ( <i>optional</i> ). Default: ";"

### Value

A dataframe object

---

run_app	<i>Run Shiny QurvE App</i>
---------	----------------------------

---

### Description

Run Shiny QurvE App

### Usage

```
run_app()
```

---

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

---

**Description**

Generic summary function for drBootSpline objects

**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**

A dataframe with statistical parameters extracted from the dose-response bootstrapping analysis.

---

summary.drFit	<i>Generic summary function for drFit objects</i>
---------------	---

---

**Description**

Generic summary function for drFit objects

**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**

A dataframe with parameters for all samples extracted from the dose-response analysis.

---

```
summary.drFitFLModel
```

*Generic summary function for drFitFLModel objects*

---

### Description

Generic summary function for drFitFLModel objects

### Usage

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

### Arguments

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

### Value

A dataframe with biosensor response parameters.

---

```
summary.drFitModel
```

*Generic summary function for drFitModel objects*

---

### Description

Generic summary function for drFitModel objects

### Usage

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

### Arguments

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

### Value

A dataframe with parameters extracted from the dose-response analysis of a single sample.



---

summary.drFitSpline	<i>Generic summary function for drFitSpline objects</i>
---------------------	---

---

**Description**

Generic summary function for drFitSpline objects

**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**

A dataframe with parameters extracted from the dose-response analysis of a single sample.

---

summary.flBootSpline	<i>Generic summary function for flBootSpline objects</i>
----------------------	--

---

**Description**

Generic summary function for flBootSpline objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with statistical parameters extracted from the dose-response bootstrapping analysis.

---

summary.flFit	<i>Generic summary function for flFit objects</i>
---------------	---

---

**Description**

Generic summary function for flFit objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with parameters extracted from all fits of a workflow.

---

summary.flFitLinear	<i>Generic summary function for flFitLinear objects</i>
---------------------	---

---

**Description**

Generic summary function for flFitLinear objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with parameters extracted from the linear fit.

---

summary.flFitSpline	<i>Generic summary function for flFitSpline objects</i>
---------------------	---

---

**Description**

Generic summary function for flFitSpline objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with parameters extracted from the nonparametric fit.

---

summary.gcBootSpline	<i>Generic summary function for gcBootSpline objects</i>
----------------------	--

---

**Description**

Generic summary function for gcBootSpline objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with statistical parameters extracted from the spline fit bootstrapping computation.

---

summary.gcFit	<i>Generic summary function for gcFit objects</i>
---------------	---

---

**Description**

Generic summary function for gcFit objects

**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.

**Value**

A dataframe with parameters extracted from all fits of a workflow.

---

summary.gcFitLinear	<i>Generic summary function for gcFitLinear objects</i>
---------------------	---

---

**Description**

Generic summary function for gcFitLinear objects

**Usage**

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

**Arguments**

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

**Value**

A dataframe with parameters extracted from the linear fit.

---

summary.gcFitModel	<i>Generic summary function for gcFitModel objects</i>
--------------------	--

---

**Description**

Generic summary function for gcFitModel objects

**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**

A dataframe with parameters extracted from the growth model fit.

---

summary.gcFitSpline	<i>Generic summary function for gcFitSpline objects</i>
---------------------	---

---

**Description**

Generic summary function for gcFitSpline objects

**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**

A dataframe with parameters extracted from the nonparametric fit.

---

`table_group_fluorescence_linear`*Generate a grouped results table for linear fits with average and standard deviations*

---

**Description**

Generate a grouped results table for linear fits with average and standard deviations

**Usage**

```
table_group_fluorescence_linear(flTable, html = FALSE)
```

**Arguments**

<code>flTable</code>	An object of class <code>flTable</code>
<code>html</code>	(Logical) Should column headers contain html formatting?

**Value**

A data frame with grouped linear fit results

---

`table_group_fluorescence_spline`*Generate a grouped results table for spline fits with average and standard deviations*

---

**Description**

Generate a grouped results table for spline fits with average and standard deviations

**Usage**

```
table_group_fluorescence_spline(flTable, html = FALSE)
```

**Arguments**

<code>flTable</code>	An object of class <code>flTable</code>
<code>html</code>	(Logical) Should column headers contain html formatting?

**Value**

A data frame with grouped spline fit results

---

`table_group_growth_linear`*Generate a grouped results table for linear fits with average and standard deviations*

---

**Description**

Generate a grouped results table for linear fits with average and standard deviations

**Usage**

```
table_group_growth_linear(gcTable, html = FALSE)
```

**Arguments**

<code>gcTable</code>	An object of class <code>gcTable</code>
<code>html</code>	(Logical) Should column headers contain html formatting?

**Value**

A data frame with grouped linear fit results

---

`table_group_growth_model`*Generate a grouped results table for parametric fits with average and standard deviations*

---

**Description**

Generate a grouped results table for parametric fits with average and standard deviations

**Usage**

```
table_group_growth_model(gcTable, html = FALSE)
```

**Arguments**

<code>gcTable</code>	An object of class <code>gcTable</code>
<code>html</code>	(Logical) Should column headers contain html formatting?

**Value**

A data frame with grouped model fit results

---



*Generate a grouped results table for spline fits with average and standard deviations*

---

### Description

Generate a grouped results table for spline fits with average and standard deviations

### Usage

```
table_group_growth_spline(gcTable, html = FALSE)
```

### Arguments

gcTable	An object of class gcTable
html	(Logical) Should column headers contain html formatting?

### Value

A data frame with grouped spline fit results

---

xgx_breaks_log10	<i>Sets the default breaks for log10 (from R package 'xgxr')</i>
------------------	--

---

### Description

xgx\_breaks\_log10 sets nice breaks for log10 scale. it's better than the default function because it ensures there is at least 2 breaks and also, it will try to go by 3s (i.e. 1,3,10,30,100) if it makes sense

### Usage

```
xgx_breaks_log10(data_range)
```

### Arguments

data_range	range of the data
------------	-------------------

### Details

for the extended breaks function, weights is a set of 4 weights for

1. simplicity - how early in the Q order are you
2. coverage - labelings that don't extend outside the data: range(data) / range(labels)
3. density (previously granularity) - how close to the number of ticks do you get (default is 5)
4. legibility - has to do with fontsize and formatting to prevent label overlap

### Value

numeric vector of breaks



**References**

Talbot, Justin, Sharon Lin, and Pat Hanrahan. "An extension of Wilkinson's algorithm for positioning tick labels on axes." *IEEE Transactions on visualization and computer graphics* 16.6 (2010): 1036-1043.

---

`xgx_minor_breaks_log10`*Sets the default minor\_breaks for log10 scales (from R package 'xgxr')*

---

**Description**

`xgx_minor_breaks_log10` sets nice `minor_breaks` for `log10` scale.

**Usage**

```
xgx_minor_breaks_log10(data_range)
```

**Arguments**

`data_range`      range of the data

**Value**

numeric vector of breaks

# Index

## \* dose-response analysis functions

flFit, 15  
growth.drBootSpline, 24  
growth.drFitSpline, 27  
growth.gcFit, 29  
growth.workflow, 38

## \* fluorescence fitting functions

flBootSpline, 14  
flFit, 15  
flFitSpline, 19

## \* growth fitting functions

growth.drFit, 25  
growth.gcBootSpline, 28  
growth.gcFit, 29  
growth.gcFitLinear, 30  
growth.gcFitModel, 34  
growth.gcFitSpline, 35  
growth.workflow, 38

## \* reports

growth.report, 37

## \* workflows

flFit, 15  
growth.gcFit, 29  
growth.workflow, 38

.onAttach, 4

AIC, 34

colFmt, 4

FALSE, 44, 76

fl.control, 4, 7–10, 14, 16, 17, 19

fl.drFit, 6, 7, 12, 14, 16, 53

fl.drFitModel, 8, 8, 48

fl.report, 9, 14

fl.workflow, 4, 10, 10, 53, 55, 58, 60, 76

flBootSpline, 6, 13, 14, 16, 17, 21, 57

flFit, 7, 14, 15, 15, 21, 25, 28, 30, 42, 55, 58, 76

flFitLinear, 6, 12, 16, 17, 19, 58

flFitSpline, 6, 12, 15–17, 19, 19, 62

grofit.param, 21

growth.control, 21, 21, 24–29, 31, 34, 35, 38, 41

growth.drBootSpline, 17, 24, 25, 26, 28, 30, 42, 45

growth.drFit, 14, 16, 23, 25, 25, 29, 30, 33, 35, 36, 41, 42, 45, 47, 53

growth.drFitModel, 26, 50

growth.drFitSpline, 17, 25, 26, 27, 30, 42, 52

growth.gcBootSpline, 23, 26, 28, 30, 33, 35, 36, 40, 42, 64

growth.gcFit, 17, 25, 26, 28, 29, 29, 33, 35, 36, 41, 42, 65, 66, 74, 76

growth.gcFitLinear, 23, 26, 29, 30, 30, 31, 35, 36, 40, 42, 65

growth.gcFitModel, 21, 26, 29, 30, 33, 34, 36, 42, 66

growth.gcFitSpline, 23, 26, 29, 30, 33, 35, 35, 40, 42, 68

growth.report, 37, 41

growth.workflow, 17, 21, 25, 26, 28–30, 33, 35–37, 38, 44, 45, 53, 65, 66, 72, 74, 76

lm, 18, 32

lowess, 27

match.arg, 42

match\_arg, 42

nls, 34

nlsLM, 8

numberInput, 42

parse\_data, 4, 11, 15, 21, 29, 39, 44, 69

plot.dr\_parameter, 52

plot.drBootSpline, 25, 45, 52

plot.drFit, 46

plot.drFitFLModel, 48

plot.drFitModel, 9, 49

plot.drFitSpline, 28, 51

plot.dual, 54

plot.flBootSpline, 56, 62

plot.flFit (plot.flFitRes), 59

plot.flFitLinear, 57

plot.flFitRes, 59

`plot.flFitSpline`, [20](#), [62](#)  
`plot.gcBootSpline`, [15](#), [28](#), [63](#), [68](#)  
`plot.gcFitLinear`, [65](#)  
`plot.gcFitModel`, [34](#), [66](#)  
`plot.gcFitSpline`, [17](#), [32](#), [36](#), [67](#)  
`plot.grodata`, [69](#)  
`plot.grofit`, [71](#)  
`plot.parameter`, [73](#)

`read_data`, [4](#), [11](#), [15](#), [21](#), [29](#), [39](#), [45](#), [55](#), [69](#), [75](#)  
`read_file`, [77](#)  
`run_app`, [77](#)

`shiny::numericInput()`, [42](#)  
`shiny::updateNumericInput()`, [43](#)  
`smooth.spline`, [6](#), [13](#), [19](#), [20](#), [23](#), [24](#), [27](#), [36](#)  
`summary.drBootSpline`, [78](#)  
`summary.drFit`, [78](#)  
`summary.drFitFLModel`, [79](#)  
`summary.drFitModel`, [79](#)  
`summary.drFitSpline`, [80](#)  
`summary.flBootSpline`, [80](#)  
`summary.flFit`, [81](#)  
`summary.flFitLinear`, [81](#)  
`summary.flFitSpline`, [82](#)  
`summary.gcBootSpline`, [82](#)  
`summary.gcFit`, [83](#)  
`summary.gcFitLinear`, [83](#)  
`summary.gcFitModel`, [84](#)  
`summary.gcFitSpline`, [84](#)

`table_group_fluorescence_linear`, [85](#)  
`table_group_fluorescence_spline`, [85](#)  
`table_group_growth_linear`, [86](#)  
`table_group_growth_model`, [86](#)  
`table_group_growth_spline`, [87](#)  
`TRUE`, [44](#), [76](#)

`validateCssUnit()`, [43](#)

`xgx_breaks_log10`, [87](#)  
`xgx_minor_breaks_log10`, [88](#)