Package 'CFAssay'

April 15, 2019

Type Package

Title Statistical analysis for the Colony Formation Assay

Version 1.16.1

Date 2017-02-10

Author Herbert Braselmann

Maintainer Herbert Braselmann <braselm@helmholtz-muenchen.de>

Depends R (>= 2.10.0)

Description The package provides functions for calculation of linear-quadratic cell survival curves and for ANOVA of experimental 2-way designs along with the colony formation assay.

License LGPL

biocViews ImmunoOncology, CellBasedAssays, CellBiology, Regression, Survival

git_url https://git.bioconductor.org/packages/CFAssay

git_branch RELEASE_3_8

git_last_commit 9d0521b

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

R topics documented:

`cellsurvLQdiff` *Comparison of two linear-quadratic cell survival curves*

Description

The function does an ANOVA test for overall comparison of the parameters $alpha$ and $beta$ of two linear-quadratic cell survival curves. The parameters are fitted simultaneously to the data with this function, i.e. no other function is necessary to derive the fits.

Usage

```
cellsurvLQdiff(X, curvevar, method="ml", PEmethod="fit")
```

Arguments

`X`	A data frame which contains columns `Exp, dose, ncells, ncolonies` and a further column containing two different values (character strings), which identify the two curves. Moreover, if there is no 0-value in the `dose`-column, `X` has to contain a column `pe` for plating efficiencies.
`curvevar`	Character string, which has to be one of the column names of the data frame `X`, that contains the two different values (character strings that distinguishes between the two curves).
`method`	Determines the method used for the fit. `"ml"` is for maximum-likelihood, `"ls"` for least-squares. `"franken"` performs weigthed least-squares with weights as described in Franken et al. (2006).
`PEmethod`	Controls the value of the plating efficiencies, i.e. the colony counts for untreated cells. `"fit"` calculates fitted plating efficiencies as model parameters, `"fix"` uses fixed ones calculated from the observed zero dose data or from a column named `pe` in `X`.

Details

In the data frame `X`, `Exp` identifies the experimental replicates and may be numeric or non-numeric. `method="ml"` for maximum-likelihood uses R function `glm` with `family` "quasipoisson" and link function `"log"`. `method="ls"` uses R function `lm`.

Value

The function returns an object of class `cellsurvLQdiff` containing three elements, `fit1`, `fit2` and `anv`. `fit1` and `fit2` are objects of class `glm` when `method="ml"` or of class `lm` when `method="ls"`. `fit1` has parameters alpha and beta fitted in common for both cell survival curves. `fit2` has parameters alpha and beta fitted differently for both curves. `anv` is of class `anova` and contains the F-test. Test results are printed, however, the full result inlcuding curve parameters is returned invisibly, i.e. the function has to be used with `print` or assigned to a variable, say for e.g. `fitcomp` as in the example below.

Author(s)

Herbert Braselmann

See Also

glm and family with references for generalized linear modelling. anova, cellsurvLQfit.

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
names(datatab) #contains a column "cline"
table(datatab$cline)
fitcomp<- cellsurvLQdiff(datatab, curvevar="cline") #using default options
print(fitcomp)
plot(cellsurvLQfit(subset(datatab, cline=="okf6TERT1")), col=1)
plot(cellsurvLQfit(subset(datatab, cline=="cal33")), col=2, add=TRUE)
legend(0, 0.02, c("okf6TERT1", "cal33"), text.col=1:2)
#using different options:
print(cellsurvLQdiff(datatab, curvevar="cline", method="ls"))
print(cellsurvLQdiff(datatab, curvevar="cline", PEmethod="fix"))
print(cellsurvLQdiff(datatab, curvevar="cline", method="ls", PEmethod="fix"))
print(cellsurvLQdiff(datatab, curvevar="cline", method="franken"))
```

---

cellsurvLQfit *Fit the linear-quadratic (LQ) model to cell survival data*

Description

This function calculates the linear coefficient $alpha$ and the coefficient $beta$ of the dose-squared term (see manual for this R-package) for colony counts measured for a set of irradiation doses and repeated experiments. The function is a wrapper for the R-functions `glm` or `lm`, which simplifies use of these functions for cell survival data.

Usage

```
cellsurvLQfit(X, method="ml", PEmethod="fit")
```

Arguments

`X`	A data frame which contains at least columns `Exp, dose, ncells, ncolonies` and if there is no 0-value in the `dose`-column, `X` has to contain a further column `pe` for plating efficiencies.
`method`	Determines the method used for the fit. `"ml"` is for maximum-likelihood, `"ls"` for least-squares. `"franken"` performs weigthed least-squares with weights as described in Franken et al. (2006).
`PEmethod`	Controls the value of the plating efficiencies, i.e. the colony counts for untreated cells. `"fit"` calculates fitted plating efficiencies as model parameters, `"fix"` uses fixed ones calculated from the observed zero dose data or from a column named `pe` in `X`.

Details

In the data frame X, Exp identifies the experimental replicates and may be numeric or non-numeric. method="ml" uses R function glm with quasipoisson family and link function "log". method="ls" uses R function lm. PEmethod="fit" fits plating efficiencies for every experiments. PEmethod="fix" uses observed plating efficiencies. If there is no 0-value in the dose-column, PEmethod is overwritten with "fix" and X has to contain a further column pe containing the plating efficiencies, i.e. ncolonies/ncells from untreated cells, not per hundred or percent.

Value

The function returns an object of class cellsurvLQfit, which is similar to classes glm or lm, however containing two additional entries, type and PEmethod, which are used for printing and plotting. The full result is returned invisibly, i.e. the function has to be used with print or plot or assigned to a variable, say for e.g. fit as in the example below.

Author(s)

Herbert Braselmann

References

Franken NAP, Rodermond HM, Stap J, et al. Clonogenic assay of cells in vitro. Nature Protoc 2006;1:2315-19.

See Also

glm and family with references for generalized linear modelling, lm

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X) #using default options
print(fit)
print(fit$type)
print(fit$PEmethod)
#Using other options:
print(cellsurvLQfit(X, method="ls"))
print(cellsurvLQfit(X, PEmethod="fix"))
print(cellsurvLQfit(X, method="ls", PEmethod="fix"))
print(cellsurvLQfit(X, method="franken"))
```

---

cfa2way *Two-way analysis for cell survival data*

Description

The function does an ANOVA of cell survival data from experimental 2-way designs where a treatment factor is tested on a control and on an altered cell line or where two different simultaneous treatments are tested on cells from a common unaltered clone. The function is a wrapper for the R-function glm. quasipoisson family is used with link function "log", i.e. dependency of treatment factors is considered as logarithmically additive.

Usage

```
cfa2way(X, A, B, param="A/B", method="ml")
```

Arguments

X	a data frame which contains columns Exp, ncells, ncolonies and two further columns for the treatment variables, see details.
A	a character string containing the name of a treatment or cell line variable (first factor in the model)
B	a character string containing the name of a treatment or cell line variable (second factor in the model)
param	Controls the parametrization of the model. Options are "A/B" for B nested in A , "B/A" for A nested in B and "A*B" for interaction term.
method	determines the method used for the fit. "ml" is for maximum-likelihood, "ls" for least-squares.

Details

In the data frame X, Exp identifies the experimental replicates and may be numeric or non-numeric. The two treatment or cell line columns should have numeric values 0, 1, ... for 2, 3, ... levels. For e.g. if a column describes clonal alteration (transfection, knock-down etc.) by a gene then 0 means unaltered or control and 1 means altered. Similar if a column describes treatment with one dose then 0 means untreated and 1 treated. 2 would indicate another dose level from the same treatment drug without taking it as a continuous covariate as for cell survival curves for radiation.

Value

The function returns an object of class cfa2way containing three elements, fit1, fit2 and anv. fit1 and fit2 are objects of class glm when method="ml" or of class lm when method="ls". fit1 has logarithmic additive parameters without interaction. fit2 has logarithmic additive parameters and interaction. anv is of class anova and contains the F-test. The full result is returned invisibly, i.e. the function has to be used with print or assigned to a variable, say for e.g. fitcomp as in the example below.

Author(s)

Herbert Braselmann

See Also

glm and family with references for generalized linear modelling.

Examples

```
datatab<- read.table(system.file("doc", "exp2_2waycfa.txt", package="CFAssay"), header=TRUE, sep="\t")
names(datatab)  # has columns "x5fuCis" and "siRNA"
fitcomp<- cfa2way(datatab, A="siRNA", B="x5fuCis", param="A/B")
print(fitcomp, labels=c(A="siRNA",B="x5fuCis"))
print(cfa2way(datatab, A="siRNA", B="x5fuCis", param="A/B", method="ls"))
```

pes *Calculation of plating efficiencies from a curve data set containing one specified curve*

Description

The function calculates plating efficiencies, i.e. fractions of colonies per cell of untreated cells, for every experimental replicate in a data frame with one specified curve. For that, lines with zero dose ($dose = 0$) are extracted from the data frame.

Usage

```
pes(X)
```

Arguments

X	A data frame which contains columns Exp, dose, ncells, ncolonies.

Details

In the data frame X, Exp identifies the experimental replicates and may be numeric or non-numeric.

Value

The function returns a data frame with three columns Exp, pe and S0, containing experiment identifiers (biological replicates), measured plating efficiencies and plating efficiencies fitted separately for each repeated experiments. Rows of the data frame are named for the different experiments.

Author(s)

Herbert Braselmann

Examples

```
datatab <- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\
X <- subset(datatab, cline=="okf6TERT1")  #Specification of curve
pes(X)
pes(subset(datatab, cline=="cal33") )
```

plot.cellsurvLQfit *Plot of an LQ model fit*

Description

This function plots a cell survival curve derived from fitting an LQ model with function cellsurvLQfit

Usage

```
## S3 method for class 'cellsurvLQfit'
plot(x, xlim = NULL, ylim = c(0.008, 1), xlab = "Dose (Gy)", ylab = "Survival (1 = 100%)",col=1, pch
```

Arguments

x	an object of class cellsurvLQfit resulting from function cellsurvLQfit.
xlim	plot range for the x-axis. Default is the dose range of the data.
ylim	plot range for the y-axis. Default is from 0.008 to 1.0
xlab	label for the x-axis. Default is "Dose (Gy)".
ylab	label for the y-axis. Default is "Survival (1 = 100%)".
col	colour for plot. Default is col = 1.
pch	symbol for plotting points. Default is pch = 1.
add	logical; if TRUE add to an already existing plot, see curve.
...	further arguments to pass to R function plot.

Author(s)

Herbert Braselmann

See Also

cellsurvLQfit

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X)
plot(fit)
S0 <- pes(X)$S0
names(S0) <- pes(X)$Exp
sfpmean(X, S0) #values of plotted mean survival fractions and error bars
# add second plot
plot(cellsurvLQfit(subset(datatab, cline=="cal33")), col=2, add=TRUE)
```

---

plotExp *Generic plotting of experimental repeats*

Description

Generic plotting of experimental repeats of cell survival data in separated plots. plotExp methods are defined for objects resulting from CFAssay functions cellsurvLQfit and cfa2way.

Usage

```
plotExp(x, ...)
```

Arguments

x	should be an object of class cellsurvLQfit or cfa2way, see details.
...	further arguments to pass to generic function plotExp, dependent of the class of the argument x.

Details

For other objects than of class `cellsurvLQfit` or `cfa2way`, `plot.default` will be called. In this version of CFAssay this will give an error message and a hint to use one of the two defined classes.

Author(s)

Herbert Braselmann

See Also

`plotExp.cellsurvLQfit` and `plotExp.cfa2way`

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X)
#pdf("okf6TERT1_experimental_plots.pdf")
   plotExp(fit)
#dev.off()
## Not run:
x <- 1
plotExp(X) #yields an error for this data type

## End(Not run)
```

---

plotExp.cellsurvLQfit *Diagnostic plots of LQ model versus experiments*

Description

This function plots the fit of an LQ model versus the fits of each experiment in a series of plots. It is recommended to direct it in a pdf-file.

Usage

```
## S3 method for class 'cellsurvLQfit'
plotExp(x, xlim = NULL, ylim = c(0.001, 1.5), xlab = "Dose (Gy)", ylab = "Survival (1 = 100%)", ...)
```

Arguments

`x`	an object of class `cellsurvLQfit` resulting from function `cellsurvLQfit`.
`xlim`	range of x axis (dose).
`ylim`	range of y-axis.
`xlab`	label of x-axis.
`ylab`	label of y-axis.
`...`	further arguments to pass to generic function `plotExp`.

Author(s)

Herbert Braselmann

See Also

cellsurvLQfit

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X)
print(fit)
#pdf("okf6TERT1_experimental_plots.pdf")
   plotExp(fit)
#dev.off()
```

---

plotExp.cfa2way *Diagnostic plot of experimental replicates for two-way analysis of cell survival data*

Description

This function plots the estimated means of a two-way ANOVA for each experiment in a series of plots. It is recommended to direct it in a pdf-file.

Usage

```
## S3 method for class 'cfa2way'
plotExp(x, labels=c(A="A",B="B"), ...)
```

Arguments

x	an R object of class cfa2way and resulting from function cfa2way.
labels	Labels for output description. These describe the meaning of A and of B.
...	further arguments to pass to generic function plotExp.

Author(s)

Herbert Braselmann

See Also

cfa2way

Examples

```
datatab<- read.table(system.file("doc", "exp2_2waycfa.txt", package="CFAssay"), header=TRUE, sep="\t")
names(datatab)  # has columns "x5fuCis" and "siRNA"
fitcomp<- cfa2way(datatab, A="siRNA", B="x5fuCis", param="A/B")
print(fitcomp, labels=c(A="siRNA",B="x5fuCis"))
  pdf("TwoWay_experimental_plots.pdf")
    plotExp(fitcomp, labels=c(A="siRNA", B="x5fuCis"))
  dev.off()
```

plotExp.default *Default function for plotting of experimental repeats*

Description

plotExp.default is the formal default method of the generic plotExp function.

Usage

```
## Default S3 method:
plotExp(x, ...)
```

Arguments

x	should be an object of class cellsurvLQfit or cfa2way, see details.
...	further arguments to pass to generic function plotExp, dependent of the class of the argument x.

Details

In this version of CFAssay other objects than of class cellsurvLQfit or cfa2way this will give an error message and a hint to use one of the two defined classes.

Author(s)

Herbert Braselmann

See Also

plotExp.cellsurvLQfit and plotExp.cfa2way

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X)
#pdf("okf6TERT1_experimental_plots.pdf")
   plotExp(fit)
#dev.off()
## Not run:
x <- 1
plotExp(X) #yields an error for this data type

## End(Not run)
```

print.cellsurvLQdiff *Print test results of comparison of two linear-quadratic cell survival curves*

Description

The function prints linear coefficients alpha and dose-squared coefficients beta of calculated with function `cellsurvLQdiff`. In addition quality statistics of the fit and ANOVA F-test results for overall comparison of the coefficients are printed.

Usage

```
## S3 method for class 'cellsurvLQdiff'
print(x, ...)
```

Arguments

`x`	an object of class `cellsurvLQdiff` resulting from function `cellsurvLQdiff`.
`...`	further arguments to pass to R function `print`.

Author(s)

Herbert Braselmann

See Also

`cellsurvLQdiff`

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
fitcomp<- cellsurvLQdiff(datatab, curvevar="cline") #using default options
print(fitcomp)
#using different options for cellsurvLQdiff:
print(cellsurvLQdiff(datatab, curvevar="cline", method="ls"))
print(cellsurvLQdiff(datatab, curvevar="cline", PEmethod="fix"))
print(cellsurvLQdiff(datatab, curvevar="cline", method="ls", PEmethod="fix"))
print(cellsurvLQdiff(datatab, curvevar="cline", method="franken"))
```

print.cellsurvLQfit *Print summary of an LQ-model fit for cell survival data*

Description

The function prints the results of an LQ-model fit for radiation dose dependent cell survival.

Usage

```
## S3 method for class 'cellsurvLQfit'
print(x, ...)
```

Arguments

x	an object of class cellsurvLQfit resulting from cellsurvLQfit.
...	further arguments to pass to R function print.

Details

In this version of CFAssay the class argument x is checked by its entry fit$type and results in an error, when x results from an independent use of glm or lm.

Author(s)

Herbert Braselmann

See Also

cellsurvLQfit

Examples

```
datatab<- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t
X<- subset(datatab, cline=="okf6TERT1")
fit<- cellsurvLQfit(X) #using default options
print(fit)
print(fit$type)
print(fit$PEmethod)
#using other options
print(cellsurvLQfit(X, method="ls"))
print(cellsurvLQfit(X, PEmethod="fix"))
print(cellsurvLQfit(X, method="ls", PEmethod="fix"))
print(cellsurvLQfit(X, method="franken"))
```

---

print.cfa2way *Print summary of two-way analysis for cell survival data.*

Description

The function prints a summary of two-way analysis for cell survival data.

Usage

```
## S3 method for class 'cfa2way'
print(x, labels=c(A="A",B="B"), ...)
```

Arguments

x	an R object of class cfa2way and resulting from function cfa2way.
labels	labels for output description. These describe the meaning of A and of B.
...	further arguments to pass to R function print.

Author(s)

Herbert Braselmann

See Also

cfa2way

Examples

```
datatab<- read.table(system.file("doc", "exp2_2waycfa.txt", package="CFAssay"), header=TRUE, sep="\t")
names(datatab)  # has columns "x5fuCis" and "siRNA"
fitcomp<- cfa2way(datatab, A="siRNA", B="x5fuCis", param="A/B")
print(fitcomp, labels=c(A="siRNA",B="x5fuCis"))
```

`sfpmean` *Pointwise mean survival fractions for curves with several experimental replicates*

Description

The function calculates mean survival fractions for curves averaged over experimental replicates. The function is employed by function `plot.cellsurvLQfit for plotting observed means`

Usage

```
sfpmean(X, S0=NULL)
```

Arguments

`X`	A data frame which contains columns `Exp, dose, ncells, ncolonies` and if `S0=NULL`, `X` has to contain a further column `pe` for plating efficiencies.
`S0`	If not `NULL`, a named numerical vector of length equal to the number of different experiments, i.e. `length(S0)==length(unique(X$Exp))` has to be `TRUE`. Default is `S0=NULL`, i.e. undefined.

Details

In the data frame `X`, `Exp` identifies the experimental replicates and may be numeric or non-numeric. `S0` may contain plating efficiencies for each replicate, resulting from function `pes` or from `cellsurvLQfit` (fitted). When `S0=NULL`, X must have a column with name `pe`, containing the plating efficiencies.

Value

A numerical matrix with two rows, the first row containing the survival fractions for each radiation dose, second row the standard deviations.

Author(s)

Herbert Braselmann

See Also

`pes`, `cellsurvLQfit`, `plot.cellsurvLQfit`

Examples

```
datatab <- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\
X <- subset(datatab, cline=="okf6TERT1")
S0 <- pes(X)$pe  #observed plating efficiencies
length(S0)==length(unique(X$Exp)) #length ok?
names(S0) <- pes(X)$Exp
sfpmean(X, S0)
fit <- cellsurvLQfit(X)
fit$coef  #contains fitted log-pe
grep("Exp",names(fit$coef))
S01 <- exp(fit$coef[1:8])  #fitted pe
sfpmean(X, S01)
## Not run: sfpmean(X)  #yields an error for this data set
```

Index