Package ‘drugCombo’

October 15, 2019

Type Package

Title Drug Interaction Modeling Based on Loewe Additivity Following Harbron’s Approach

Version 1.1.1

Date 2019-10-15

Author Maxim Nazarov, Nele Goeyvaerts, Chris Harbron (original article and code)

Maintainer Maxim Nazarov <maxim.nazarov@openanalytics.eu>

Description Perform assessment of synergy/antagonism for drug combinations based on the Loewe additivity model, following Harbron's approach (Statistics in Medicine, 2010, <doi:10.1002/sim.3916>). The package allows flexible modeling of the drug interaction index and supports “2-stage” estimation in addition to “1-stage” estimation, including bootstrap-based confidence intervals. The method requires data on the monotherapy responses and the package accommodates both checkerboard and ray designs. Functions are available for graphical exploration of model goodness-of-fit and diagnostics, as well as for synergy/antagonism assessment in 2D and 3D.

License GPL-3

RoxygenNote 6.1.1

Depends R (>= 2.10)

Imports ggplot2, minpack.lm, BIGL (>= 0.9), rgl, nlme, Deriv, methods, stats

Suggests knitr, rmarkdown, testthat, Cairo

VignetteBuilder knitr, rmarkdown

LazyData true

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-15 14:50:06 UTC
### Description

Show estimated model parameters from all bootstrap iterations

### Usage

```r
bootstrapCoefs(tauSurface)
```

### Arguments

- `tauSurface` A `tauSurface` object returned by `getTauSurface`.

### Value

A matrix of parameter estimates from the bootstrap iterations as returned by `getBootTaus`.

### Author(s)

Maxim Nazarov
checkerboardData

Example checkerboard design drug combination data

Description
A dataset containing data from 10 different in-vitro drug combination experiments for an antiviral treatment (compound 1) using a checkerboard design. The variables are as follows:

Format
A data frame with 9360 observations of 5 variables:
- d1: dose of the first compound
- d2: dose of the second compound
- effect: observed effect (normalized cell counts)
- plate: plate ID (1 to 3 within experiment)
- exp: experiment ID (1 to 10)

contour.tauSurface

Contour-plot method for "tauSurface" objects

Description
Contour-plot method for "tauSurface" objects

Usage
## S3 method for class 'tauSurface'
contour(x, digits = 4, ...)

Arguments
- x: A tauSurface object returned by getTauSurface.
- digits: Number of digits used in format for the dose labels.
- ...: Further parameters, currently not used.

Details
The function returns a heatmap-like plot displaying the interaction index (tau) estimates and point-wise confidence intervals numerically. Blue and red colours are used to indicate areas of synergy (if confidence interval lies entirely below 1) or antagonism (if confidence interval lies entirely above 1). Note that color intensity is determined by the absolute values of the interaction indices.
Value
a ggplot2 object

Author(s)
Maxim Nazarov

---

**fitModel**

*Fit drug interaction index model according to Harbron’s framework*

**Description**

This is the main function to fit an interaction index model to drug combination data based on the Loewe additivity model. The interaction index can be specified in a flexible way as a function of doses and other variables.

**Usage**

```r
fitModel(data, mono = NULL, model = NULL, tauFormula = NULL, tauLog = FALSE, tauStart = 1 * (!tauLog), stage = 1, fixed = NULL, inactiveIn = 0, verbose = FALSE, ...)
```

**Arguments**

- **data** A (long) data frame to fit the model to. Required columns are "d1", "d2" and "effect". Other variables are allowed and can be used in `tauFormula`.
- **mono** An optional "MarginalFit" object, obtained from `fitMarginals`.
- **model** A pre-defined model to use for the interaction index tau. One of "additive", "uniform", "linear1", "separate1", "linear2", "separate2", "separate12" or "zhao". See details.
- **tauFormula** A formula to define the interaction index tau, using either 'literal' (as in `nls`) or 'symbolic' (as in `lm`) specification.
- **tauLog** Whether to fit the model using log-transformed tau parameters. This is mostly useful for "separate"-type tau models for better convergence. Note that if TRUE, tau cannot be negative, which may be not appropriate for some models, such as "linear1" and "linear2". Note that this affects the coefficient names in the result ("logtau1", "logtau2", ... instead of "tau1", "tau2", ...), so if `fixed` argument is used, this should be taken into account.
- **tauStart** Vector of starting values for tau parameters, either of length 1 or of the same length as the total number of tau parameters.
- **stage** Whether to run a 1-stage or 2-stage estimation.
- **fixed** Constraints on monotherapy and/or tau parameters as a vector of the form 'name = value', if NULL (default), taken from `mono`. Note that the tau parameters should be named "tau1", "tau2", ... if `tauLog = FALSE` (default), or "logtau1", "logtau2", ... if `tauLog = TRUE`.
inactiveIn  which compound is inactive (1 or 2), or 0 (default) when both compounds are active.

verbose  Whether to show extra information useful for debugging.

...  Further arguments passed to the nlsLM call. For example, trace = TRUE is useful to see the trace of the fitting process and may help identify issues with convergence.

Details

There are different ways to specify a model for the interaction index tau:

- Using one of the pre-defined models as specified in the model argument:
  - "additive", for additivity model,
  - "uniform", one overall value for tau,
  - "linear1", linear dependency on log10 dose of the first compound,
  - "linear2", linear dependency on log10 dose of the second compound,
  - "separate1", different tau for each dose of the first compound,
  - "separate2", different tau for each dose of the second compound,
  - "separate12", different tau for each combination of doses of the two compounds,
  - "zhao", quadratic response surface model following Zhao et al 2012.

- Using a literal or symbolic formula. Note that for the monotherapies, tau is assumed to be equal to 1. Therefore, continuous models may entail discontinuities in the interaction index when d1 and d2 approach 0.

Value

Fitted object of class "HarbronFit" which is an nls-like object with extra elements.

Author(s)

Maxim Nazarov

Examples

data("checkerboardData", package = "drugCombo")
data1 <- checkerboardData[checkerboardData$exp == 1, ]
mono1 <- fitMarginals(data1, fixed = c(b = 1))
# all three ways below are equivalent
fitLin1 <- fitModel(data = data1, mono = mono1, model = "linear1")
fitLin1b <- fitModel(data1, mono1, tauFormula = ~ log10(d1))
fitLin1c <- fitModel(data1, mono1, tauFormula = ~ tau1+tau2*log10(d1))
fitPlot2d

Plot 2d surface (slices) of observations and model fit

Description
Plot 2d surface (slices) of observations and model fit

Usage
fitPlot2d(fit, fit2 = NULL, side = c("d1", "d2", "total"),
useFineGrid = TRUE, modelNames = NULL)

Arguments
fit A HarbronFit object returned by fitModel.
fit2 An optional HarbronFit object returned by fitModel. If provided, a 2d-plot comparing two model fits is produced. Note that the two models should have been fitted on the same data. Note that this argument can also be used as side.
side Which side ("d1", "d2" or "total" for the sum of d1 and d2) to use as x-axis.
useFineGrid Whether to use fine grid for plotting fitted curves (default), or calculate predictions only at the observed data points.
modelNames Model names to use for the plot legend in the case of model comparison (i.e. when fit2 is provided).

Value
ggplot2 object

Author(s)
Maxim Nazarov

fitPlot3d

Plot 3d surface of observations and model fit

Description
Plot 3d surface of observations and model fit

Usage
fitPlot3d(fit, logScale = TRUE, useFineGrid = TRUE, showMesh = TRUE, widget = FALSE)
**getBootTaus**

**Arguments**

- **fit**
  A HarbronFit object returned by `fitModel`.

- **logScale**
  Whether to use log-scale for x and y axes.

- **useFineGrid**
  Whether to use fine grid for plotting the fitted surface (default), or calculate predictions only at the observed data points.

- **showMesh**
  Whether to show 'mesh' on the plot. Note: currently doesn’t play nicely with the useFineGrid argument, so the latter is ignored.

- **widget**
  Whether to return a "htmlwidget" object instead of plotting on a 3d device.

**Value**

A 3d plot is shown or an object of class "htmlwidget" as returned by `rglwidget`.

**Author(s)**

Maxim Nazarov

---

**getBootTaus**

Run nonparametric bootstrap on the interaction index model

**Description**

Function to run nonparametric bootstrap on the interaction index model. It is usually called from `getTauSurface`.

**Usage**

```r
getBootTaus(fit, niter = 100, resample = c("all", "mono", "stratified"), seed = NULL, verbose = FALSE, ...)
```

**Arguments**

- **fit**
  A HarbronFit object returned by `fitModel`.

- **niter**
  Number of bootstrap samples to use.

- **resample**
  Resampling method for bootstrap. Either "all" (default) for resampling from all data, "mono" for separately resampling monotherapy and combination data, or "stratified" for resampling at each dose combination separately. Note that the latter method is not meaningful if there are no replicates in the data.

- **seed**
  Random seed to use for bootstrap

- **verbose**
  Whether to show progress information.

- **...**
  Further arguments passed to the `fitModel` calls.

**Value**

A matrix of interaction index estimates based on the bootstrap samples.
getTauSurface

**Description**

Computes estimates and confidence intervals for the interaction surface for all dose combinations.

**Usage**

```r
getTauSurface(fit, data = NULL, addCI = TRUE, method = c("default", "boot"), level = 0.95, niter = 100, resample = c("all", "mono", "stratified"), seed = NULL, ...)
```

**Arguments**

- `fit`: A HarbronFit object returned by `fitModel`.
- `data`: Dose combinations to compute interaction index for. If NULL (default), taken from the `fit` object.
- `addCI`: Whether confidence intervals need to be computed.
- `method`: Which method to use to calculate confidence intervals: "default" for Wald-type or "boot" for non-parametric bootstrap.
- `level`: The confidence level required for the confidence intervals (default is 0.95).
- `niter`: Number of bootstrap samples to use.
- `resample`: Resampling method for bootstrap. Either "all" (default) for resampling from all data, "mono" for separately resampling monotherapy and combination data, or "stratified" for resampling at each dose combination separately. Note that the latter method is not meaningful if there are no replicates in the data.
- `seed`: Random seed to use for bootstrap.
- `...`: Further parameters that are passed to `getBootTaus`.

**Value**

An object of class "tauSurface" which is essentially a list with the following components: data frame with interaction index (tau) estimates, standard errors and point-wise confidence intervals, formulas for computing tau at any given dose (only for models with continuous functions used to define tau), and details on the tau specification from the `fit`. In addition, if the "boot" method was used, all the bootstrap estimates are returned and can be accessed with `bootstrapCoefs`.

**Author(s)**

Maxim Nazarov
invL4

See Also
plot.tauSurface, contour.tauSurface for visual representation of the tau surface.

invL4

Inverse 4-parameter log-logistic (response-dose) function

Description
Inverse 4-parameter log-logistic (response-dose) function

Usage
invL4(y, h, b, m, logEC50)

Arguments
y Response.

h Hill’s coefficient (slope of the curve).

b Baseline effect (at zero dose).

m Maximal/asymptote effect (at infinite dose).

logEC50 Point of inflection (in logarithmic terms).

Value
Dose level.

plot.HarbronFit

Plot method for HarbronFit objects

Description
Diagnostic and goodness-of-fit plots (in 2D and 3D).

Usage
## S3 method for class 'HarbronFit'
plot(x, y = NULL, which = c("nls", "2d", "3d"),
     ...)
Arguments

- **x**: A HarbronsFit object returned by `fitModel`.
- **y**: A (optional) second HarbronsFit object returned by `fitModel`. If provided, which argument is ignored and a 2d-plot comparing two model fits is produced. Note that the two models should have been fitted on the same data. Note that this argument can also be used as `which`. See examples.
- **which**: Whether to show default `plot.nls`("nls"), a 'slice' plot with fitted curves overlaid on top of the observed data ("2d") or a 3d-plot with fitted surface overlaid on top of the observed data ("3d").
- **...**: Further arguments passed to `plot.nls`, `fitPlot2d` or `fitPlot3d` depending on which.

Value

Output from `plot.nls`, a ggplot2 object if `which = "2d"`, or a 3d rgl plot if `which = "3d"`.

Author(s)

Maxim Nazarov

Examples

```r
data("checkerboardData", package = "drugCombo")
data1 <- checkerboardData[checkerboardData$exp == 1, ]
fitUniform <- fitModel(data1, model = "uniform")
fitLinear <- fitModel(data1, model = "linear1")
plot(fitUniform, fitLinear)
plot(fitLinear, "2d") # here 2nd argument is interpreted as 'which'
```

---

### plot.tauSurface

**Plot method for "tauSurface" objects**

**Description**

2D slice and 3D surface plots for the interaction index surface.

**Usage**

```r
## S3 method for class 'tauSurface'
plot(x, y = NULL, which = c("2d", "3d"), ...)
```
Arguments

- **x**: A `tauSurface` object returned by `getTauSurface`.
- **y**: An optional second `tauSurface` object returned by `getTauSurface`. If provided, which argument is ignored and a 2d-plot comparing two tau surfaces is produced. Note that the two estimates should have been calculated on the same `HarbronFit` object. Note that this argument can also be used as `which`. See examples.
- **which**: Whether to show a 3d plot (surface plot) or a 2d plot (slice plot). Correspondingly `tauPlot3d` or `tauPlot2d` is called.
- **...**: Further arguments passed to `tauPlot3d` or `tauPlot2d`.

See Also

- `tauPlot3d`, `tauPlot2d` for the underlying functions and their arguments.
- `contour.tauSurface` for another visual representation of the estimated interaction indices.

Examples

```r
data("checkerboardData", package = "drugCombo")
data1 <- checkerboardData[checkerboardData$exp == 1, ]
fitUniform <- fitModel(data1, model = "uniform")
tauUniform <- getTauSurface(fitUniform)
fitLinear <- fitModel(data1, model = "linear1")
tauLinear <- getTauSurface(fitLinear)
plot(tauUniform)
plot(tauLinear, which = "2d", side = "d2", facetBy = "d1")
plot(tauLinear, which = "3d")
plot(tauUniform, tauLinear, tauNames = c("uniform", "linear"))
plot(tauUniform, tauLinear, continuous2 = FALSE)
```
rayData

Example ray design drug combination data

Description

A dataset containing data from an in-vitro drug combination experiment in oncology using a ray design. The variables are as follows:

Format

A data frame with 378 observations of 4 variables:

- d1: dose of the first compound
- d2: dose of the second compound
- effect: observed effect (radioactivity level)
- ray: a character vector with values 0, 0.2, 0.35, 0.5, 0.65, 0.8 and 1, corresponding to the mixture factors. Each ray has 9 dose combinations with 6 replicates.

tauPlot2d

Plot estimated interaction index surface slice along one of the doses

Description

Plot estimated interaction index surface slice along one of the doses

Usage

tauPlot2d(tauSurface, tauSurface2 = NULL, side = "d1", 
groupBy = NULL, colorBy = groupBy, facetBy = "auto", 
continuous = FALSE, continuous2 = FALSE, addLine = continuous || 
continuous2, funs = NULL, funs2 = NULL, title = NULL, 
tauNames = NULL, digits = 4, facetOpts = NULL)

Arguments

tauSurface A tauSurface object returned by getTauSurface.
tauSurface2 An optional second tauSurface object returned by getTauSurface. If provided, a 2d-plot comparing the two tau surfaces is produced. Note that the estimates should have been calculated on the same HarbronFit object.
side Data column to use as x-axis: "d1", "d2", "total" or another variable from the data in the tauSurface object.
groupBy Data column to use as grouping. Note that if comparison of two surfaces is performed, this will be ignored.
colorBy Data column to use for coloring.
tauPlot2d

facetBy
Whether to facet plots by extra variables used in the tau formula ("auto") or manually provided data column(s) to facet by.

continuous
Whether continuous type of plot is requested (for tauSurface1). This is automatically detected if used via plot.tauSurface, but can be overwritten.

continuous2
Whether continuous type of plot is requested (for tauSurface2). This is automatically detected if used via plot.tauSurface, but can be overwritten.

addLine
Whether to connect tau estimates for subsequent doses.

funs
A list with functions to compute tau surface and confidence bands (for tauSurface1). These are returned by the getTauSurface and are automatically used when the plot.tauSurface is called.

funs2
A list with functions to compute tau surface and confidence bands (for tauSurface2). These are returned by the getTauSurface and are automatically used when the plot.tauSurface is called.

title
Plot title.

tauNames
Tau surface names to use for the plot legend in the case of comparison of estimates (i.e. when tauSurface2 is provided).

digits
Number of digits used in format for the dose labels.

facetOpts
Arguments passed to facet_wrap

Details

The function returns a 2d plot for the interaction index (tau) estimates as a function of one of the two doses in a checkerboard design, or rays in a ray design. Pointwise confidence intervals are displayed as error bars. In addition to plotting tau estimates from one tauSurface object, the function can be used to compare two tauSurface objects. This can be used, for example, to see the difference between Wald-type and bootstrap-based confidence intervals for tau. Although the function can be used 'manually', typically one calls the plot.tauSurface method, which then calls this function when which = "2d".

Value

A ggplot2 object.

Author(s)

Maxim Nazarov

See Also

plot.tauSurface, tauPlot3d
tauPlot3d

Plot 3d surface of interaction index estimates

Description

Plot 3d surface of interaction index estimates

Usage

tauPlot3d(tauSurface, logScale = TRUE, continuous = FALSE,
        funs = NULL, addPlane = continuous, colorPoints = TRUE,
        widget = FALSE)

Arguments

tauSurface  A tauSurface object returned by getTauSurface.
logScale     Whether to use log-scale for x and y axes.
continuous  Whether continuous type of plot is requested. This is automatically detected if
             used via plot.tauSurface, but can be overwritten.
funs        A list with functions to compute tau surface and confidence bands. These are re-
             turned by the getTauSurface and are automatically used when the plot.tauSurface
             is called.
addPlane     Whether to add estimated tau plane.
colorPoints  Whether to color points by synergy/antagonism. Blue color is used for points
             deemed synergistic (if the confidence interval lies entirely below 1), red for
             points deemed antagonistic’ (if the confidence interval lies entirely above 1). Other
             points are colored white.
widget       Whether to return a "htmlwidget" object instead of plotting on 3d device.

Details

The function returns a 3d plot for the interaction index (tau) estimates as a function of the doses of
the two drugs. Pointwise confidence intervals are displayed as error bars. Although the function
can be used 'manually', typically one calls the plot.tauSurface method, which then calls this
function when which = "3d".

Value

A 3d plot is shown or an object of the class "htmlwidget" as returned by rglwidget.

Author(s)

Maxim Nazarov

See Also

plot.tauSurface, tauPlot2d
**unique.tauSurface**

Unique method for "tauSurface" objects

---

**Description**

"unique" method to extract unique interaction index estimates from a "tauSurface" object in a tabular format.

**Usage**

```
## S3 method for class 'tauSurface'
unique(x, ...)
```

**Arguments**

- `x` Output of `getTauSurface`.
- `...` Further arguments, currently not used.
Index

bootstrapCoefs, 2, 8
checkerboardData, 3
contour.tauSurface, 3, 9, 11
facet_wrap, 13
fitMarginals, 4
fitModel, 4, 6–8, 10
fitPlot2d, 6, 10
fitPlot3d, 6, 10
format, 3, 13

getBootTaus, 2, 7, 8
getTauSurface, 2, 3, 7, 8, 11–15

invL4, 9
nlsLM, 5

plot.HarbronFit, 9
plot.nls, 10
plot.tauSurface, 9, 10, 13, 14
print.tauSurface, 11

rayData, 12
rglwidget, 7, 14

tauPlot2d, 11, 12, 14
tauPlot3d, 11, 13, 14

unique.tauSurface, 15