Package ‘LoBrA’

May 7, 2026

Type Package

Title Generalized Spline Mixed Effect Models for Longitudinal Breath Data

Version 1.0

Date 2022-03-07

Description Automated analysis and modeling of longitudinal 'omics' data (e.g. breath 'metabolomics') using generalized spline mixed effect models. Including automated filtering of noise parameters and determination of breakpoints.

Depends R (>= 3.5.0)

LazyData true

License GPL-3

RoxygenNote 7.1.2

Encoding UTF-8

Imports methods, lawstat, nlme, graphics, RColorBrewer, stats, base, grDevices, qpdf

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Anne-Christin Hauschild [aut, cre] (ORCID:
<`https://orcid.org/0000-0002-7499-4373`>),
Sandy P. Eckel [ths, com],
Jan Baumbach [ths]

Maintainer Anne-Christin Hauschild <`achauschild@googlemail.com`>

Repository CRAN

Date/Publication 2022-03-21 18:40:04 UTC

Contents

`as.LOBdataset` *Transformation of a single longitudinal data matrix into 'LoBrA' Data Object.*

Description

Real signals and background noise originating from experimental settings or random events

Usage

```
as.LOBdataset(
  longData,
  name = "",
  id = "id",
  time = "time",
  type = "type",
  class = "class",
  bg = FALSE
)
```

Arguments

`longData`	Matrix of longitudinal data containing all components
`name`	name of the dataset
`id`	name to identify the experiment id column
`time`	name to identify the time column
`type`	name to identify the type column
`class`	name to identify the class column
`bg`	indicates whether the data table contains background data

Value

'LoBrA' data object

Examples

```
## Not run:

  data(LoBraExample)
  name="Longitudinal Test Dataset"
  ldo<-as.LOBdataset(longDataExample, name, bg=TRUE)
```

`components` *'LoBrA' Data Object (LDO) for Example data set*

Description

'LoBrA' example LDO created by the function 'createExampleData' and converted to an LDO by 'as.LOBdataset' function. It consist of a single matrix for all experiments, time points, types (background, experiment), class and the intensity values of all components created. The artificial data consist of 20 experiments and 100 components with 18 measurements (3 background, 15 sample). The 10 experiments are each associated to on of 2 groups (ONE and TWO). The components comprise 70 noise components and 30 components that randomly vary in their trajectories in one of three segments. Random noise is added to all intercepts, propagated and added to each time point for all samples and components separately.

Usage

```
components
```

Format

A vector of selected components from the longitudinal example data set.

Author(s)

Anne-Christin Hauschild `<hauschild@uni-marburg.de>`

`createBGComponents` *Simulate background noise peaks*

Description

Simulating background noise signals originating from experimental settings or random events

Usage

```
createBGComponents(
  components,
  samples,
  labels,
  timepoints = 15,
  bg = 3,
  mean = 5,
  sd = 3,
  experimentSD = 2,
  randomnoise = 0.1,
  plotting = FALSE
)
```

Arguments

`components`	number of background components to be created
`samples`	number of experiments
`labels`	name of each experiment
`timepoints`	number of sample measurements
`bg`	number of background measurements
`mean`	mean value of noise components
`sd`	standard deviation value of noise for this component
`experimentSD`	standard deviation value of each experiment for this component
`randomnoise`	random variation changing at each time point
`plotting`	Indicator whether the component will be plotted (TRUE) or not (FALSE)

Value

matrix of background components

`createBGData` *Simulate background measurements*

Description

Simulating background noise signals originating from experimental settings or random events

Usage

```
createBGData(samples = 10, bg = 3, mean = 0, sd = 1, randomnoise = 0.1)
```

Arguments

`samples`	number of experiments
`bg`	number of background measurements
`mean`	mean value of noise for this component
`sd`	standard deviation value of noise for this component
`randomnoise`	random variation changing at each time point

Value

matrix of background measurements

`createExampleData` *Create example data set for 'LoBrA'*

Description

Real signals and background noise originating from experimental settings or random events

Usage

```
createExampleData(
  components = c(70, 10, 10, 10),
  samples = 10,
  classes = 2,
  bg = 3,
  timepoints = rep(5, 3),
  myfile = NA
)
```

Arguments

`components`	vector numbers of background and informative components to be created.
`samples`	number of experiments per class
`classes`	number of classes
`bg`	number of background measurements
`timepoints`	number of sample measurements for each spline
`myfile`	filename of the pdf file created. Note: '.pdf' is added automatically.

Value

final matrix of example data.

Examples

```
## Not run:

  components = c(21,3,3,3)
  samples = 10
  classes = 2;
  bg = 3;
  timepoints = rep(5,3)
  p=TRUE;
  longDataExample <- createExampleData(components, samples, classes, bg,
                                       timepoints)
  dim(longDataExample)
```

`createGoudermanData` *Create the Gouderman Data Arrangement.*

Description

Using the Gouderman methodology to create the Gouderman-Data Arrangement.

Usage

```
createGoudermanData(selectedLDO, breaks, center, timeperiod = NA, range = NA)
```

Arguments

`selectedLDO`	Longitudinal Data Object, containing all selected metabolites to be used for the final Gouderman model.
`breaks`	break points for the spline model
`center`	Time point that corresponds to the center time t0. The algorithm will test whether there is a significant difference between the groups at this point.

`timeperiod`	If the user defines the time period or segment, in the spline to be tested. Note, a 3 break point spline has 4 segments.
`range`	If the user defines a range, the algorithm will test whether there is a significant difference between the groups in this range.

Value

The function returns a 'GaudermanLDO' object. For more information @seealso 'GaudermanLDO' .

Examples

```
## Not run:

  data(LoBraExample)
  selectedLDO <- selectComponents(ldo, components)
  breaks<- c(8, 12)
  center<- 12
  timeperiod <- 2;
  gaudermanLDOexample <- createGoudermanData(selectedLDO, breaks, center, timeperiod)
```

---

createInformativeComponents

Simulate informative peaks

Description

This function simulates signals correlated to different informative events.

Usage

```
createInformativeComponents(
  components,
  samples,
  labels,
  timepoints = c(5, 5, 5),
  bg = 3,
  mean = 5,
  sd = 3,
  segment = 1,
  slopeSD = 2,
  randomnoise = 0.5,
  plotting = FALSE
)
```

Arguments

`components`	number of background components to be created
`samples`	number of experiments
`labels`	label of each experiment
`timepoints`	number of sample measurements
`bg`	number of background measurements
`mean`	mean value of noise for the intercept of this components
`sd`	standard deviation value of noise for the intercept of this component
`segment`	indicating the segment, that will have an informative event (changing slope for one class)
`slopeSD`	standard deviation value for the generated slope of for this component (mean is zero, therefore, the slope can be either negative or positive)
`randomnoise`	random variation changing at each time point
`plotting`	logical value, (default is FALSE), if TRUE the function will plot the created time series.

Value

matrix of informative components

`GaudermanLDO-class` *An S4 class to represent a 'Gouderman' LDO object, that was generated by the generalized gauderman algorithm.*

Description

An S4 class to represent a 'Gouderman' LDO object, that was generated by the generalized gauderman algorithm.

Slots

- `name` character Name of the new 'generalized-Gauderman' adjusted longitudinal data
- `dataFrames` list List of 'generalized-Gauderman' modified data. One `data.frame` for each component.
- `peaknames` character Vector of component names contained in this object.
- `k` numeric Updated times for the breaks of the spline model.
- `times` matrix Vector of updated time values.
- `newTimeVars` character The names of the newly defined time variables of the generalized 'Gauderman' model.
- `ids` character Vector of identifiers for the experiments
- `labels` factor Vector of class labels for each experiment

`GaudermanModelEvaluation-class`

An S4 class to represent the result of the linear mixed effect modeling on a gauderman LDO.

Description

An S4 class to represent the result of the linear mixed effect modeling on a gauderman LDO.

Slots

- `name` character Name of the new 'generalized-Gauderman' adjusted longitudinal model.
- `gaudermanLDO` GaudermanLDO 'Generalized-Gauderman' adjusted longitudinal data object.
- `models` list List of models generated for each component.
- `labels` factor Vector of class labels for each experiment
- `pvalues` matrix Matrix of p-values for the intercept as well as all slops of the spline model for each component.
- `correctedpvalues` matrix Matrix of corrected p-values for the intercept as well as all slops of the spline model for each component.
- `modelparameter` matrix Model parameter for each component.

`getColor` *Get colors for the plotting function.*

Description

Get colors for the plotting function.

Usage

```
getColor(label, size)
```

Arguments

`label`	class labels of the samples
`size`	size of the color vector to be created

Value

col vector of colors created

---

getGeneralizedGaudermanDataFrame

Create Peak Matrices for Generalized 'Gauderman' linear mixed effect regression (LMER) Model with parameterized Times

---

Description

Create Peak Matrices for Generalized 'Gauderman' linear mixed effect regression (LMER) Model with parameterized Times

Usage

```
getGeneralizedGaudermanDataFrame(
  peakmatrix,
  sampleIds,
  classes,
  center,
  timeperiod,
  gaudermanRange,
  k
)
```

Arguments

`peakmatrix`	Peak matrix to be converted.
`sampleIds`	Ids of samples in the matrix
`classes`	Classes of samples
`center`	Time point that corresponds to the center time t0. The algorithm will test whether there is a significant difference between the groups at this point.
`timeperiod`	defines the time period or segment, in the spline to be tested. Note, a 3 break point spline has 4 segments.
`gaudermanRange`	range to be tested for a significant difference between the groups.
`k`	break points for the generalized 'Gauderman' spline model.

Value

Return the new peak matrix data frame for this peak.

`getOptimalSpline` *Extract the optimal spline model parameters from the ModelSelection Object.*

Description

The method calculates which spline model and parameters worked best with respect to the median of the specified quality measure. The median is calculated among all component models.

Usage

```
getOptimalSpline(
  lobraModelSelectionObject,
  qualityMeasure = "AIC",
  summeryfun = stats::median
)
```

Arguments

`lobraModelSelectionObject`	LDOmodelselection created by the 'lobraModelSelection' function. It stores all evaluated Spline models to chose from.
`qualityMeasure`	Quality measure to be used to select the optimal spline.
`summeryfun`	Define the Summery function to be used. Default value is set to stats::median. Other possible functions would be mean, for instance.

Value

The function returns a 'lobraModelSelectionObject' that contains the optimal model according to the specified quality measure. @seealso plot.modelSelectionEvaluation

Examples

```
## Not run:

  data(LoBraExample)
  selectedLDO <- selectComponents(ldo, components)
  potentialBreaks=c(8, 12)
  nknots=c(1, 2)
  qualityMeasure=c("AIC", "BIC")
  ldoSelect<- lobraModelSelection(selectedLDO, potentialBreaks, nknots, qualityMeasure)

 optimalAIC<-getOptimalSpline(ldoSelect, qualityMeasure="AIC", summeryfun=stats::median)
  message(optimalAIC@breaks);

  optimalBIC<-getOptimalSpline(ldoSelect, qualityMeasure="BIC", summeryfun=base::mean)
  hist(unlist(optimalBIC@quality));
```

`getPvalue` *Testing differences of groups with respect to a specific value and test.*

Description

Testing differences of groups with respect to a specific value and test.

Usage

```
getPvalue(y, group, test)
```

Arguments

`y`	Values to be tested
`group`	corresponding groups whose difference we want to test
`test`	specific test to be used. Can be each of the following 'bf', 'levene' or 'bartlett'.

`ldo` *'LoBrA' Data Object (LDO) for Example data set*

Description

'LoBrA' example LDO created by the function 'createExampleData' and converted to an LDO by 'as.LOBdataset' function. It consist of a single matrix for all experiments, time points, types (background, experiment), class and the intensity values of all components created. The artificial data consist of 20 experiments and 100 components with 18 measurements (3 background, 15 sample). The 10 experiments are each associated to on of 2 groups (ONE and TWO). The components comprise 70 noise components and 30 components that randomly vary in their trajectories in one of three segments. Random noise is added to all intercepts, propagated and added to each time point for all samples and components separately.

Usage

```
ldo
```

Format

A matrix representing 20 experiments. It contains values for 100 variables at 18 time points for each experiment. Object of class `LDO`.

Author(s)

Anne-Christin Hauschild `<hauschild@uni-marburg.de>`

`LDO-class` *An S4 class to represent a 'LoBrA' Data Object (LDO). It stores multiple time series data for muliple experiements and multiple Components. It allows repeated measurements of a component, irregular sampling, and unequal temporal spacing of the time points.*

Description

An S4 class to represent a 'LoBrA' Data Object (LDO). It stores multiple time series data for muliple experiements and multiple Components. It allows repeated measurements of a component, irregular sampling, and unequal temporal spacing of the time points.

Slots

`name` character Name of the 'LDO' object

`dataMatrices` list List of matrices of component measurement. It contains a measurement for each time point and each experiment.

`backgroundMatrices` list List of matrices of background measurements. It contains a measurement for each time point and each experiment.

`peaknames` character Character vector of Component names

`times` numeric Vector of times for each time point in the data

`ids` character Vector of identifiers for the experiments

`labels` factor Vector of class labels for each experiment

`LDOmodelselection-class` *An S4 class to represent a model selection result based on an 'LDO'.*

Description

An S4 class to represent a model selection result based on an 'LDO'.

Slots

`ldo` LDO 'LDO' object the model selection is based on.

`potentialBreaks` numeric Vector of numeric values that were considered as potential break points in the model selection.

`splinetype` character Type of spline used.

`qualityMeasure` character Quality measures used during the model selection ('AIC', 'BIC' or 'LogLik')

`modelList` list List of all models tested.

`quality` list List of quality matrices, one matrix for each quality measure used. Each matrix contains the quality for each spline tested for each component.

`breaks` list For each tested spline, this list contains a vector of breaks.

`LDOscreening-class` *An S4 class to represent the screening of metabolites in an 'LDO'.*

Description

An S4 class to represent the screening of metabolites in an 'LDO'.

Slots

`ldo` LDO LDO object the screening is based on.

`experimentIntercept` list List of experiment intercepts.

`experimentResiduals` list List of experiment Residuals.

`interceptPvalues` matrix Matrix of experiment intercept p-values.

`residualPvalues` matrix Matrix of experiment Residual p-values.

`selectedPeaks` matrix Matrix of logical values. Each entry indicates whether a specific component is significant according to a specific test.

`LoBrA` *LoBrA: A package for modeling longitudinal breath data*

Description

The `LoBrA` package provides important data objects and functions to analyze longitudinal metabolomic (breath) data.

Introduction

Novel metabolomic technologies paved the way for longitudinal analysis of exhaled air and online monitoring of fast progressing diseases. This package implements an automated analysis approach of longitudinal data from different omics technologies, such as ion mobility spectrometry of human exhaled air and demonstrates how including temporal signals increases the statistical power in biomarker identification. It can handel multiple irregular 4D time series data. More precisely, it can simultaniously handel the data of multiple experiements each observing multiple components. Therefore, it allows repeated measurements of a component, irregular sampling, and unequal temporal spacing of the time points.

LoBrA Analysis

A typical LoBrA analysis is will comprise the following steps

1. Background Screening: Using the function `screening` and `selectComponents` to select the Components that most likely do not originate from background noise.

2. Model Selection: First, a set of spline models based on different number of splits and split positions are generated by the function `lobraModelSelection`. Subsequently, these models are

evaluated using different quality criteria, i.e. 'AIC', 'BIC' and 'LogLik'. Finally, the most appropriate model is selected.

3. Evaluation of the non-background components on the selected model, using the longitudinal 'Gouderman' linear mixed effect model in function `modelGoudermanLongitudinal`.

Author(s)

Maintainer: Anne-Christin Hauschild [Copyright holder]

Authors:

- Sandrah P. Eckel
- Jan Baumbach

`lobraModelSelection` *Evaluation of different spline variants.*

Description

The model selection method evaluates which spline models achieve the best quality among all tested metabolites.

Usage

```
lobraModelSelection(
  selectedLDO,
  potentialBreaks = c(),
  nknots = c(0, 1, 2),
  splinetype = "linear",
  qualityMeasure = c("AIC", "BIC", "logLik")
)
```

Arguments

`selectedLDO`	`LDO` containing all selected metabolites to be used for the model selection.
`potentialBreaks`	Vector of all possible knots to be used for the spline modeling.
`nknots`	Vector of number of spline knots to be used. Therefore, 0 ~ no spline, 1 ~ spline with one knot, 2 ~ spline with two knots, etc.
`splinetype`	spline type default is 'linear'. (Currently only linear is supported.)
`qualityMeasure`	Vector of quality measures to be used. Possible options are 'AIC', 'BIC', and 'logLik'.

Value

`LDOmodelselection` Object. For each quality measure the model list contains a list of models for each spline tested. Additionally, the output contains a matrix of qualities for each Spline Component pair. And finally there is a list of breaks for each spline tested.

Examples

```
## Not run:

  data(LoBraExample)
  potentialBreaks <- c(8,12)
  selectedLDO <- selectComponents(ldo, components)
  ldoSelect<- lobraModelSelection(selectedLDO, potentialBreaks, nknots=c( 1, 2))
  length(ldoSelect@ldo@peaknames)
```

`longDataExample` *'LoBrA' Example Data Set*

Description

'LoBrA' example data set created by the function 'createExampleData'. #' It consist of a single matrix for all experiments, time points, types (background, experiment), class and the intensity values of all components created. The artificial data consist of 20 experiments and 100 components with 18 measurements (3 background, 15 sample). The 10 experiments are each associated to on of 2 groups (ONE and TWO). The components comprise 70 noise components and 30 components that randomly vary in their trajectories in one of three segments. Random noise is added to all intercepts, propagated and added to each time point for all samples and components separately.

Usage

```
longDataExample
```

Format

A matrix representing 20 experiments. It contains values for 100 variables at 18 time points for each experiment.

id Experiment identifier

time Time Point of Measurement

type Type of Measurement (e.g. Background, or Sample measurement for each experiment)

class Class or Group id of the sample/ experiment

bgcomponent-x 70 random variables that represent the background noise of the experiments

components-x-x 30 components that randomly vary in their trajectories in one of three time periods, (1:4-8, 2:9-13, 3:14-18). ...

Author(s)

Anne-Christin Hauschild `<hauschild@uni-marburg.de>`

`modelGoudermanLongitudinal` *Fitting the Gouderman LME Model with using Gouderman-Data Arrangement.*

Description

Uses the linear mixed effects modeling to build the final 'Gauderman' model. The 'Gauderman' modification enables the exact calculation of the significance of a specified section of the spline model.

Usage

```
modelGoudermanLongitudinal(mygaudermanLDO, correctionMethod = "bonferroni")
```

Arguments

`mygaudermanLDO`	GaudermanLDO data object, created by the generalized 'Gauderman' algorithm (GGA).
`correctionMethod`	correction for p-values. Possible methods: 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'BY', 'fdr', 'none'

Value

'GaudermanModelEvaluation' Results of the evaluation of the Fitted linear mixed effect models for the defined time periods.

Examples

```
 data(LoBraExample)
 selectedLDO <- selectComponents(ldo, components)
gaudermanLDOexample <- createGoudermanData(selectedLDO, breaks=c(8, 12), center=12, timeperiod=2)
 evalResult<- modelGoudermanLongitudinal(gaudermanLDOexample)
 message(evalResult@correctedpvalues<0.005)
```

`plotGaudermanModel` *Plotting helper function to plot a single generalized gouderman Model*

Description

Plotting helper function to plot a single generalized gouderman Model

Usage

```
plotGaudermanModel(
  data,
  labels,
  ul,
  tempmodel,
  colores,
  maincol,
  breaks,
  main,
  ylab,
  xlab
)
```

Arguments

`data`	data matrix used to fit the model
`labels`	class labels for all samples
`ul`	unique class labels
`tempmodel`	model to be plotted
`colores`	predefined colors for the single samples
`maincol`	predefined colors for the fitted spline
`breaks`	break points of the spline to be plotted
`main`	main title of the plot
`ylab`	y label of the plot
`xlab`	x label of the plot

`plotGoudermanLongitudinalResults`

Plotting the 'Gouderman' LME Model and Results.

Description

Plotting the 'Gouderman' LME Model and Results.

Usage

```
plotGoudermanLongitudinalResults(
  evaluationresult,
  main = "Mixed Effect Spline Model Evaluation",
  ylab = "Value",
  xlab = "Time",
  peaknames = NULL
)
```

Arguments

`evaluationresult`	'GaudermanModelEvaluation' data object, created by the modelGouderman-Longitudinal function.
`main`	title of the plot
`ylab`	y axis label
`xlab`	x axis label
`peaknames`	selection of peaks to be plotted

Value

No return value

Examples

```
 wd <- tempdir()
 data(LoBraExample)
 selectedLDO <- selectComponents(ldo, components)
gaudermanLDOexample <- createGoudermanData(selectedLDO, breaks=c(8, 12), center=12, timeperiod=2)
 evalResult<- modelGoudermanLongitudinal(gaudermanLDOexample)
 # Plot all peaks
 filename<- file.path(wd, "finalModelEvaluation.pdf") ;
 oldpar <- par("mfrow")
 grDevices::pdf(filename, width=16, height=8);
   graphics::par(mfrow=c(1,1));
   plotGoudermanLongitudinalResults(evalResult);
 par(mfrow = oldpar)
 grDevices::dev.off();

 #Plot a selection of Peaks
 peaknames<- evalResult@gaudermanLDO@peaknames;
 filename<- file.path(wd, "finalModelEvaluation-components.pdf") ;
 oldpar <- par("mfrow")
 grDevices::pdf(filename, width=20, height=8);
   graphics::par(mfrow=c(2,5));
   plotGoudermanLongitudinalResults(evalResult, main="", peaknames=peaknames);
 par(mfrow = oldpar)
 grDevices::dev.off();
```

`plotLDOScreening` *Plotting the screening results.*

Description

For each peak two box plots are created. The first plot shows a boxplot of the Sample Intercept Comparison of the sample and the background, and the corresponding p-values. The second plot shows a boxplot of the Residual Comparison of the sample and the background, and the corresponding p-values.

Usage

```
plotLDOScreening(
  ldoscreen,
  plotAll = FALSE,
  correctionmethod = "levene",
  decs = 3,
  ask = FALSE,
  peaknames = rownames(ldoscreen@selectedPeaks)
)
```

Arguments

`ldoscreen`	LDO screening result
`plotAll`	Select all components to be plotted. Default plots only the selected peaks using the correction method.
`correctionmethod`	Version of correction method to be used to select the peaks. Valid values are 'bf', 'levene', and 'bartlett'.
`decs`	decimal numbers of p-values to be plotted.
`ask`	logical. Modifies the graphical parameter `ask` in `par` (If TRUE (and the R session is interactive) the user is asked for input, before a new figure is drawn. As this applies to the device, it also affects output by packages grid and lattice. It can be set even on non-screen devices but may have no effect there.)
`peaknames`	Defining a list of peaks to be plotted. By default all peaks will be plotted.

Value

No return value

Examples

```
## Not run:


  wd <- tempdir()
  data(LoBraExample)
  ldos<-screening(ldo, method= c('levene'), alpha =0.05, criteria=c(1,1))
  filename<- file.path(wd, "screeningresults.pdf")
  grDevices::pdf(filename, width=16, height=8)
  plotLDOScreening(ldos)
  grDevices::dev.off();
```

`plotmodelSelectionEvaluation` *Plotting results of Model Evaluation and Selection.*

Description

Plotting the results of Model Evaluation and Selection. The plot shows a vertical boxplot for each spline tested starting with the best average fit according to the selected quality measure. The label of each spline can be found on the left, the median quality measure on the right. The x-axis denotes the selected quality measure.

Usage

```
plotmodelSelectionEvaluation(
  lobraModelSelectionObject,
  qualityMeasure,
  title = NULL
)
```

Arguments

`lobraModelSelectionObject`	Object of type LDOmodelselection that was created during the model evaluation. @seealso 'lobraModelSelection'
`qualityMeasure`	List of quality measures to be visualized.
`title`	Title of the plot.

Value

No return value

Examples

```
## Not run:


wd <- tempdir()
data(LoBraExample)
selectedLDO <- selectComponents(ldo, components)
ldoSelect<- lobraModelSelection(selectedLDO, potentialBreaks=c(8, 12), nknots=c(1, 2))

filename<- file.path(wd, "evaluateBestSplineAIC.pdf") ;
grDevices::pdf(filename, width=16, height=8);
  plotmodelSelectionEvaluation(ldoSelect, "AIC", "Best Spline Models");
grDevices::dev.off();

qualityMeasure=c("AIC", "BIC", "logLik")
filename<- file.path(wd, "evaluateBestSplineAllMeasures.pdf") ;
```

```
grDevices::pdf(filename, width=16, height=8);
oldpar <- par("mfrow")
par(mfrow=c(3,1))
  plotmodelSelectionEvaluation(ldoSelect, qualityMeasure);
par(mfrow = oldpar)
grDevices::dev.off();
```

`plotTimeSeries` *Plotting function for a longitudinal data matrix (Internal Function)*

Description

Plotting function for a longitudinal data matrix (Internal Function)

Usage

```
plotTimeSeries(
  myMatrix,
  main = "",
  labels = NA,
  ylab = "Expression",
  xlab = "Time Point",
  legend = "",
  col = 1:dim(myMatrix)[1]
)
```

Arguments

`myMatrix`	longitudinal data matrix to be plotted
`main`	Title of the plot
`labels`	class labels of samples
`ylab`	Label of y axis
`xlab`	Label of x axis
`legend`	of plot
`col`	vector of colors for plot

Value

No return value

`powerSet` *Creating the power set of a set.*

Description

Creating the power set of a set.

Usage

```
powerSet(set)
```

Arguments

`set`	Set of numbers of potential spline break points.

Value

Returns power set of the given set.

`screening` *Screening of background or confounding components*

Description

Background noise signals originating from experimental settings or random events can hugely influence the signal pattern of the breath. Background data enables the detailed evaluation and differentiation of the compounds originating primarily from the background or confounding factors as compared to those from the sample itself. The method assumes that all compounds of interest show a larger variation in the sample as compared to the background noise.

Usage

```
screening(
  ldo,
  method = c("bf", "levene", "bartlett"),
  alpha = 0.05,
  criteria = c(1, 1)
)
```

Arguments

`ldo`	Longitudinal Data Object
`method`	list of tests to perform, standard values: 'bf', 'levene' or 'bartlett'). 'bf' relates to "Brown-Forsythe" Levene-type procedure, 'levene' uses classical "Levene's" procedure and 'bartlett' applies Bartlett's test.
`alpha`	A numeric value to defining the cutoff to select peaks.
`criteria`	indicators which criteria to use for screening decision.

Value

Returns an object of type 'LDOscreening' containing the original 'ldo' object and the results of the screening. The variable 'selectedPeaks' contains a matrix including the results (TRUE = Significant, FALSE = not Significant) of the specified tests ('bf', 'levene', 'bartlett').

Examples

```
## Not run:


  data(LoBraExample)
  method= c('bf', 'levene', 'bartlett')
  alpha =0.05
  criteria=c(1,1)
  ldos<-screening(ldo, method, alpha, criteria)
  components <- ldos@selectedPeaks[,"levene"]
  components <- names(components)[components]
  selectedLDO <- selectComponents(ldo, components)
```

`selectComponents` *Create a new 'LDO' Object that only contains the selected components.*

Description

Create a new 'LDO' Object that only contains the selected components.

Usage

```
selectComponents(ldo, components, name = paste(ldo@name, " selected"))
```

Arguments

`ldo`	Longitudinal Data Object
`components`	Component names to select for the new ldo object. Only elements from this list that overlap with the peak names in the given ldo, are utilized.
`name`	Name of newly created 'LDO' object.

Value

new ldo object only containing the selected components.

Index