Package: mrbin (via r-universe)

September 10, 2024

Title Metabolomics Data Analysis Functions

Version 1.7.3

Description A collection of functions for processing and analyzing metabolite data. The namesake function mrbin() converts 1D or 2D Nuclear Magnetic Resonance data into a matrix of values suitable for further data analysis and performs basic processing steps in a reproducible way. Negative values, a common issue in such data, can be replaced by positive values (<doi:10.1021/acs.jproteome.0c00684>). All used parameters are stored in a readable text file and can be restored from that file to enable exact reproduction of the data at a later time. The function fia() ranks features according to their impact on classifier models, especially artificial neural network models.

Imports grDevices, graphics, stats, utils, methods

Depends R (>= 2.10)

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Suggests parallel, keras

VignetteBuilder utils

URL http://www.kleinomicslab.com/software/, https://github.com/kleinomicslab/mrbin

Repository https://kleinomicslab.r-universe.dev

RemoteUrl https://github.com/kleinomicslab/mrbin

RemoteRef HEAD

RemoteSha 4a67ad256db16991d35a0e7982ec8ab91904d99e

Contents

`addToPlot` *A function for adding NMR spectra to the plot list.*

Description

This function adds a spectrum to the plot list.

Usage

```
addToPlot(
  folder = NULL,
  dimension = "1D",
  NMRvendor = "Bruker",
  useAsNames = "Folder names",
  add = TRUE
)
```

Arguments

`folder`	Defines the exact NMR data folder. If NULL, mrbin parameter set is used
`dimension`	Defines the data dimension, "1D" or "2D". Only used if not NULL
`NMRvendor`	Defines the NMR manufacturer, default is "Bruker"
`useAsNames`	How should sample names be generated
`add`	Add spectra to existing list, or replace existing spectra. Default is TRUE

Value

none

Examples

```
addToPlot()
```

`annotatemrbin` *A function for annotating mrbin objects.*

Description

This function annotates an mrbin object and returns it with updated $annotations vector

Usage

```
annotatemrbin(mrbinObject)
```

Arguments

mrbinObject	An mrbin object

Value

An (invisible) mrbin object

Examples

```
 mrbinObject<-mrbin(silent=TRUE,
                    parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
                binwidth1D=0.04,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,0.5,10,156),
                    saveFiles="No",referenceScaling="No",noiseRemoval="No",
                    fixNegatives="No",logTrafo="No",noiseThreshold=.05,tryParallel=FALSE,
                    NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                               system.file("extdata/3/10/pdata/10",package="mrbin"))),
                    metadata=list(metaboliteIdentities=matrix(c(
                    1.346,1.324,21,23,
                    3.052,3.043,30.5,33.5,
                    4.066,4.059,57,59.5
                    ),ncol=4,byrow=TRUE,dimnames=list(
                    c("Lactate","Creatinine","Creatinine"),NULL))))
mrbinObject<-annotatemrbin(mrbinObject)
mrbinObject$metadata$annotations[125:135]
plotPCA(mrbinObject,loadings=TRUE)
```

---

atnv *A function replacing negative values.*

Description

This function replaces (column-wise) negative values by a small positive number. The number is calculated as an affine transformation to the range of the lowest positive number to 0,01*the lowest positive number (of this column). Ranks stay unchanged. Positive numbers are not altered. If sample-wise noise levels are available, the median noise level of samples with negative values is calculated and replaces the lowest positive number in case it is smaller. If no noise data is available, the 1 positive values in the data set is used as an estimate. It is recommended to us this function AFTER noise removal and other data clean-up methods, as it may alter (reduce) the noise level.

Usage

```
atnv(NMRdata, noiseLevels = NULL, verbose = TRUE, errorsAsWarnings = FALSE)
```

Arguments

NMRdata	A matrix or mrbin object containing NMR data. For matrix: columns=frequencies,rows=samples
noiseLevels	A vector (can be omitted if NMRdata is an mrbin object)
verbose	Should a summary be displayed if NMRdata is an mrbin object

errorsAsWarnings
: If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible matrix or mrbin object containing NMR data without negative values.

Examples

```
resetEnv()
Example<-mrbin(silent=TRUE,
                 parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
            binwidth1D=0.005,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,7.5,10,156),
                 saveFiles="No",referenceScaling="No",noiseRemoval="No",
                 fixNegatives="No",logTrafo="No",noiseThreshold=.05,
                 NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                            system.file("extdata/3/10/pdata/10",package="mrbin"))
                 ))
sum(Example$bins<=0)
exampleNMRpositive<-atnv(NMRdata=Example$bins, noiseLevels=Example$parameters$noise_level_adjusted)
sum(exampleNMRpositive<=0)
```

---

checkmrbin *A function for checking mrbin objects.*

Description

This function checks an mrbin object and returns warning if changes were not documented

Usage

```
checkmrbin(mrbinObject, verbose = TRUE, errorsAsWarnings = NULL)
```

Arguments

mrbinObject
: An mrbin object

verbose
: Should a summary be displayed? (Warnings will be displayed even when setting verbose to FALSE)

errorsAsWarnings
: If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data. If not provided, this will be taken from the mrbinObject.

Value

An (invisible) character vector of warnings

Examples

```
mrbinObject<-createmrbin()
mrbinObject<-checkmrbin(mrbinObject)
```

`contMin` *A function for changing plotNMR plots.*

Description

This function decreases the minimum contour level of the current 2D NMR spectrum plot.

Usage

```
contMin(refreshPlot = TRUE)
```

Arguments

`refreshPlot`	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
mrbin(silent=TRUE,parameters=list(dimension="2D",binwidth2D=0.5,
         binheight=3,PQNScaling="No",referenceScaling="No",binRegion=c(4,3,60,65),
         noiseRemoval="No",trimZeros="No",cropHSQC="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
         NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"))))
plotNMR()
contMin()
```

`contPlus` *A function for changing plotNMR plots.*

Description

This function increases the minimum contour level of the current 2D NMR spectrum plot.

Usage

```
contPlus(refreshPlot = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
addToPlot(folder=system.file("extdata/1/12/pdata/10",package="mrbin"),dimension="2D")
plotNMR()
contPlus()
```

---

createmrbin *A function for creating mrbin objects.*

Description

This function creates an mrbin object and returns it.

Usage

```
createmrbin()
```

Value

An (invisible) mrbin object

Examples

```
mrbinObject<-createmrbin()
```

---

cropNMR *A function for cropping HSQC spectra.*

Description

This function crops HSQC spectra to the region along the diagonal to remove uninformative signals. Will work only for 1H-13C HSQC spectra.

Usage

```
cropNMR()
```

Value

None

Examples

```
resetEnv()
Example<-mrbin(silent=TRUE,
         parameters=list(dimension="2D",binwidth2D=1,binheight=4,cropHSQC="No",PCA="No",
      PQNScaling="No",noiseRemoval="No",removeSolvent="No",verbose=TRUE,tryParallel=FALSE,
         NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"))))
cropNMR()
```

---

dilutionCorrection *A function for scaling to individual dilution factors.*

Description

This function performs sample-wise scaling of binned data to correct for dilution through different sample volumes used, or for different sample weights. All bin values of one sample are multiplied by the corresponding dilution factor.

Usage

```
dilutionCorrection(mrbinResults, verbose = TRUE, errorsAsWarnings = FALSE)
```

Arguments

mrbinResults	An mrbin object
verbose	Should a summary be printed?
errorsAsWarnings	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible mrbin object containing scaled NMR data.

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
         binwidth1D=0.05,PQNScaling="No",PCA="No",tryParallel=FALSE,logTrafo="No",
         NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                      system.file("extdata/2/10/pdata/10",package="mrbin"),
                      system.file("extdata/3/10/pdata/10",package="mrbin"))),
         metadata=list(dilutionFactors=c(.75,1,.5)))
mrbinResults<-dilutionCorrection(mrbinResults)
```

down *A function for changing plotNMR plots.*

Description

This function moves down the plot region of the current NMR plot (only 2D).

Usage

```
down(refreshPlot = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
mrbin(silent=TRUE,parameters=list(dimension="2D",binwidth2D=0.5,
         binheight=3,PQNScaling="No",referenceScaling="No",binRegion=c(4,3,60,65),
         noiseRemoval="No",trimZeros="No",cropHSQC="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
         NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"))))
plotNMR()
zoomIn()
down()
```

editmetabolitesmrbin *A function for editing metabolite identities.*

Description

This function edits the metabolite list within an mrbin object and returns it

Usage

```
editmetabolitesmrbin(mrbinObject, borders, metabolitenames, add = TRUE)
```

Arguments

mrbinObject	An mrbin object
borders	A matrix of signal borders. 1D: two columns: left, right 2D: four columns: left, right, top, bottom
metabolitenames	A character vector of metabolite identities
add	Should the new metabolite list be added to an existing list, or replace the current list?

Value

An (invisible) mrbin object

Examples

```
 mrbinObject<-mrbin(silent=TRUE,
                    parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
               binwidth1D=0.04,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,0.5,10,156),
                    saveFiles="No",referenceScaling="No",noiseRemoval="No",
                    fixNegatives="No",logTrafo="No",noiseThreshold=.05,tryParallel=FALSE,
                    NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                               system.file("extdata/3/10/pdata/10",package="mrbin"))
                    ))
 mrbinObject<-editmetabolitesmrbin(mrbinObject,borders=matrix(c(
     1.346,1.324,
     3.052,3.043,
     4.066,4.059
   ),ncol=2,byrow=TRUE),metabolitenames=c(
   "Lactate",
   "Creatinine",
   "Creatinine"
   ))
mrbinObject$parameters$metaboliteIdentities
```

---

editmrbin *A function for editing mrbin objects.*

Description

This function edits an mrbin object and returns it. This is the only documented way to edit mrbin objects, all other ways of editing such object might cause warning message

Usage

```
editmrbin(
  mrbinObject,
  functionName = "mrbin::editmrbin",
```

```
  versionNumber = as.character(utils::packageVersion("mrbin")),
  bins = NULL,
  parameters = NULL,
  metadata = NULL,
  transformations = NULL,
  comment = "",
  verbose = TRUE
)
```

Arguments

mrbinObject	An mrbin object
functionName	Name of the package and function calling this command
versionNumber	Version number of the package calling this command
bins	A matrix containing values to be written to the mrbin object
parameters	A list containing values to be written to the mrbin object parameters, names must be names of the mrbin object, e.g. dimension
metadata	A list containing values to be written to the mrbin object parameters, names must be names of the mrbin object
transformations	An optional character vector describing any used data transformations or scaling such as reference scaling, PQN, log, atnv, etc.
comment	An optional character vector describing the change
verbose	Should a summary be displayed?

Value

An (invisible) mrbin object

Examples

```
mrbinObject<-createmrbin()
mrbinObject<-editmrbin(mrbinObject)
```

---

fia *A function identifying features of importance.*

Description

This function finds features that can change the outcomes of a model's prediction. Example: fia=1.00 means single compound found in all but 0 percent of samples. fia=2.45 indicates this compound is found in pairs in all but 45 percent of tested samples A function named predict needs to be present for this to work. If the function name of the prediction function is different, the function name has to be provided in the parameter functionNamePredict.

Usage

```
fia(
  model,
  dataSet,
  factors,
  nSeed = 6,
  numberOfSamples = 100,
  maxFeatures = 10000,
  innerLoop = 300,
  verbose = TRUE,
  maxNumberAllTests = 5,
  firstLevel = 1,
  saveMemory = FALSE,
  kerasClearMemory = 0,
  functionNamePredict = "predict",
  parameterNameObject = "object",
  parameterNameData = "x",
  ...
)
```

Arguments

model	A predictive model. Make sure to have loaded all required packages before starting this function
dataSet	An object containing data, columns=features, rows=samples. This should be either a matrix or a dataframe, depending on which of these two the specific prediction function requires
factors	A factor vector with group membership of each sample in the data set. Order of levels must correspond to the number predicted by the model
nSeed	Number of times that the test will be repeated, selecting different random features
numberOfSamples	Number of samples that will be randomly chosen from each group
maxFeatures	Maximum number of features that will be tested. Larger numbers will be split into child nodes without testing to increase speed
innerLoop	Number of repeated loops to test additional child nodes
verbose	A logical vector to turn messages on or off
maxNumberAllTests	Combinations of features of this length or shorter will not be split in half to create two children, but into multiple children with one feature left out each. This is done make sure no combination is missed.
firstLevel	Numeric value of first level or group. Usually 1 but for glm such as in the example this needs to be 0.
saveMemory	Save memory by performing only two predictions per step, which will be much slower. If you are using keras, use parameter kerasClearMemory=2 instead to free more memory and be a lot faster. FALSE to turn off.

kerasClearMemory
: Save memory by clearing model from memory and reloading the model between chunks of predictions. Will only work when using package keras. 0=off, 1=medium (reload between repeat with different seeds), 2=maximum memory savings (reload after each run for a single sample). This will write a model file to the working directory.

functionNamePredict
: The name of the prediction function. This only needs to be changed if the prediction function is not called predict

parameterNameObject
: The name of the parameter for passing the model to the prediction function

parameterNameData
: The name of the parameter for passing the data to the prediction function

...
: Optional, additional parameters that will be passed to the prediction function.

Value

A list of results: scoresSummary A vector of fia scores for the whole dataset; scores contains vectors of fia scores for each predicted group; scoresIndividual A list of fia scores for each individual sample; fiaListPerSample A list of important combinations of features for each predicted sample; fiaMatrix A list of fia scores for each predicted group.

Examples

```
#First, define group membership and create the example feature data
group<-factor(c(rep("Group1",4),rep("Group2",5)))
names(group)<-paste("Sample",1:9,sep="")
dataset<-data.frame(
  Feature1=c(5.1,5.0,6.0,2.9,4.8,4.6,4.9,3.8,5.1),
  Feature2=c(2.6,4.0,3.2,1.2,3.1,2.1,4.5,6.1,1.3),
  Feature3=c(3.1,6.1,5.8,5.1,3.8,6.1,3.4,4.0,4.4),
  Feature4=c(5.3,5.2,3.1,2.7,3.2,2.8,5.9,5.8,3.1),
  Feature5=c(3.2,4.4,4.8,4.9,6.0,3.6,6.1,3.9,3.5),
  Feature6=c(6.8,6.7,7.2,7.0,7.3,7.1,7.2,6.9,6.8)
  )
rownames(dataset)<-names(group)
#train a model - here we use a logit model instead of ANN as a demonstration
mod<-glm(group~Feature1+Feature2+Feature3+Feature4+Feature5+Feature6,
  data=data.frame(group=group,dataset),family="binomial")
fiaresults<-fia(model=mod,dataSet=dataset,factors=group,parameterNameData="newdata",
  firstLevel=0,type="response")
fiaresults$scores
```

---

getEnv *A function for saving the package environment.*

Description

This function returns a list of all objects of the current package environment. This may be helpful for debugging or for accessing NMR spectral data and the raw bin data.

Usage

```
getEnv()
```

Value

A list containing all objects from the local package environment.

Examples

```
tempList<-getEnv()
```

---

intMin *A function for changing plotNMR plots.*

Description

This function decreases the intensity of the current NMR spectrum plot.

Usage

```
intMin(dimension = "1D", refreshPlot = TRUE, value = NULL)
```

Arguments

dimension	Dimension to use. Defaults to "1D"
refreshPlot	Refresh plot automatically. Defaults to TRUE
value	Set exact value. Defaults to NULL

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
         PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
intMin()
```

intPlus *A function for changing plotNMR plots.*

Description

This function increases the intensity of the current NMR spectrum plot.

Usage

```
intPlus(dimension = "1D", refreshPlot = TRUE)
```

Arguments

dimension	Dimension to use. Defaults to "1D"
refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
         PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
intPlus()
```

left *A function for changing plotNMR plots.*

Description

This function moves left the plot region of the current NMR plot.

Usage

```
left(refreshPlot = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
mrbin(silent=TRUE,parameters=list(dimension="1D",binwidth1D=.5,
         noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         PQNScaling="No",saveFiles="No",referenceScaling="No",
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
zoomIn()
left()
```

`logTrafo` *A function for log transforming data.*

Description

This function performs logarithm transformation. Will not work with negative data.

Usage

```
logTrafo(mrbinResults, verbose = TRUE, errorsAsWarnings = FALSE)
```

Arguments

`mrbinResults`	An mrbin object
`verbose`	Should a summary be printed?
`errorsAsWarnings`	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible mrbin object

Examples

```
resetEnv()
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D", logTrafo="No",
             binwidth1D=0.05,signal_to_noise1D=50,verbose=TRUE,PCA="No",tryParallel=FALSE,
                   NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                                system.file("extdata/2/10/pdata/10",package="mrbin"))))
mrbinResults<-logTrafo(mrbinResults)
```

metadatamrbin *A function for interactively editing metadata of mrbin objects.*

Description

This function edits interactively or non-interactively the metadata filed of the provided mrbin object.

Usage

```
metadatamrbin(mrbinResults, metadata = NULL)
```

Arguments

mrbinResults	An mrbin object
metadata	An optional list of objects to be changed. If provided, interactive mode is deactivated

Value

An invisible mrbin object

Examples

```
mrbinObject<-mrbin(silent=TRUE,
                   parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
               binwidth1D=0.04,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,0.5,10,156),
                   saveFiles="No",referenceScaling="No",noiseRemoval="No",
                   fixNegatives="No",logTrafo="No",noiseThreshold=.05,tryParallel=FALSE,
                   NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                              system.file("extdata/3/10/pdata/10",package="mrbin"))
                   ))
mrbinObject<-metadatamrbin(mrbinObject,metadata=list(projectTitle="Test project"))
```

mrbin *A function setting the parameters and performing binning and data processing*

Description

This function guides the user through the set-up of parameters, starts binning and performs the chosen data processing steps. If a list of parameters is provided and silent is set to TRUE, no user input is requested and binning and data processing are performed silently.

Usage

```
mrbin(silent = FALSE, setDefault = FALSE, parameters = NULL, metadata = NULL)
```

Arguments

silent	If TRUE, the user will be asked no questions and binning and data analysis will run according to the current parameters. Defaults to FALSE.
setDefault	If TRUE, all current parameters will be replaced by the default parameters (before loading any provided parameters sets). Defaults to FALSE.
parameters	Optional: A list of parameters, see examples for details. If omitted, the user will be asked through a series of question to set the parameters.
metadata	Optional: A list of metadata. If omitted, the user can add metadata after generating bin data.

Value

An invisible object of type "mrbin" containing bins (data after processing), parameters, and factors

Examples

```
# Set parameters in command line.
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(
                dimension="1D",binwidth1D=0.01,tryParallel=FALSE,
                signal_to_noise1D=25,noiseThreshold=0.75,
                NMRfolders=c(
                system.file("extdata/1/10/pdata/10",package="mrbin"),
                system.file("extdata/2/10/pdata/10",package="mrbin"),
                system.file("extdata/3/10/pdata/10",package="mrbin"))
                ))
```

---

mrbinrun *A function performing all data read and processing steps.*

Description

This function reads parameters from the global variable mrbin.env$mrbin$parameters and performs the following operations: Reading NMR files, creating bins, removing solvent area, removing additional user-defined areas, summing up bins that contain unstable peaks such as citric acid, removes noise bins, crops HSQC spectra to the diagonal area, performs PQN scaling, replaces negative values, log transforms and displays a PCA plot. Parameters are then saved in a text file. These can be recreated using recreatemrbin().

Usage

```
mrbinrun(createbins = TRUE, process = TRUE, mrbinResults = NULL, silent = TRUE)
```

Arguments

createbins	If TRUE, new bin data is generated
process	If TRUE, bin data is processed, e.g. by noise removal, atnv, etc.
mrbinResults	An mrbin object. Needs to be provided only if createbins is FALSE
silent	If set to FALSE, no new time calculation is performed

Value

An invisible mrbin object

Examples

```
resetEnv()
setParam(parameters=list(dimension="2D",binwidth2D=0.1,binheight=5,
   binRegion=c(8,1,15,140),PQNScaling="No",tryParallel=FALSE,
   fixNegatives="No",logTrafo="No",signal_to_noise2D=10,solventRegion=c(5.5,4.2),
   NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"),
                system.file("extdata/2/12/pdata/10",package="mrbin"))))
mrbinrun()
```

---

mrplot *A function for plotting NMR spectra.*

Description

This function plots NMR spectra. A menu of commands is displayed to edit the plot view and add spectra. Multiple spectra will be overlaid, and if both 1D and 2D spectra are selected, they are shown in two plots with matched ranges.

Usage

```
mrplot(
  hideMenu = FALSE,
  folders = NULL,
  dimensions = NULL,
  intensity1D = NULL,
  zoom = NULL
)
```

Arguments

hideMenu	Do not show the menu. Defaults to FALSE
folders	Optional vector of folder names of spectra to load. Defaults to NULL
dimensions	Optional vector dimensions of spectra to load. Defaults to NULL
intensity1D	Optional value of initial 1D intensity. Defaults to NULL
zoom	Optional vector of initial zoom area. Defaults to NULL

Value

None

Examples

```
resetEnv()
mrplot(folders=c(system.file("extdata/1/12/pdata/10",package="mrbin"),
                 system.file("extdata/1/10/pdata/10",package="mrbin"),
                 system.file("extdata/2/10/pdata/10",package="mrbin"),
                 system.file("extdata/3/10/pdata/10",package="mrbin")),
       dimensions=c("2D","1D","1D","1D"),zoom=c(2.8,2.3,20,55),
       intensity1D=32,hideMenu=TRUE)
```

`plotMultiNMR` *A function for plotting NMR spectra.*

Description

This function plots the current NMR spectrum. If no parameters are provided, parameters are read from the mrbin.env environment variables, set by mrbin. To change the plot, use zoom(), zoomIn(), zoomOut(), intPlus(), intMin(), left(), right(). For 2D data use additionally: contMin(), contPlus(), up(), down()

Usage

```
plotMultiNMR(
  region = NULL,
  rectangleRegions = NULL,
 rectangleColors = c("green", "orange", "blue", "red", "yellow", "gray", "purple"),
  rectangleFront = FALSE,
  polygonRegion = NULL,
  maxPlots = Inf,
  color = NULL,
  add = FALSE,
  showGrid = FALSE,
  manualScale = TRUE,
  plotTitle = "",
  restrictToRange = FALSE,
  enableSplit = TRUE,
  dimension = NULL
)
```

Arguments

`region`	A vector defining the plot region (left, right, top, bottom)
`rectangleRegions`	A 4-column matrix defining areas where to plot rectangles
`rectangleColors`	Define colors for the rectangles
`rectangleFront`	Plot rectangles in front of spectrum rather than in background (only 2D)

polygonRegion	Defines 4 corners of a polygon to be plotted
maxPlots	The maximum number of 2D plots to be overlaid
color	Defines the color of the spectrum plot. If NULL, a rainbow theme is used for 2D NMR
add	If TRUE, additional spectrum plots are overlaid with the current plot
showGrid	Shows a grid of data points. Defaults to FALSE
manualScale	If TRUE, scaling factor is taken from environment variables
plotTitle	Defines the main title of the plot
restrictToRange	Restrict plot area to range of available data points. Defaults to FALSE
enableSplit	Allow split plots for showing 1D and 2D spectra simultaneously
dimension	If not provided, this will be taken from package environment

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
         PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotMultiNMR()
```

---

plotNMR *A function for plotting NMR spectra.*

Description

This function plots the current NMR spectrum. If no parameters are provided, parameters are read from the mrbin.env environment variables, set by mrbin. To change the plot, use zoom(), zoomIn(), zoomOut(), intPlus(), intMin(), left(), right(). For 2D data use additionally: contMin(), contPlus(), up(), down()

Usage

```
plotNMR(
  region = NULL,
  rectangleRegions = NULL,
 rectangleColors = c("green", "orange", "blue", "red", "yellow", "gray", "purple"),
  rectangleFront = FALSE,
  polygonRegion = NULL,
  color = NULL,
  add = FALSE,
```

```
  showGrid = FALSE,
  manualScale = TRUE,
  plotTitle = "",
  restrictToRange = FALSE,
  currentSpectrumOriginal = NULL,
  perspective = FALSE,
  noise = NULL,
  dimension = NULL
)
```

Arguments

region	A vector defining the plot region (left, right, top, bottom)
rectangleRegions	A 4-column matrix defining areas where to plot rectangles
rectangleColors	Define colors for the rectangles
rectangleFront	Plot rectangles in front of spectrum rather than in background (only 2D)
polygonRegion	Defines 4 corners of a polygon to be plotted
color	Defines the color of the spectrum plot. If NULL, a rainbow theme is used for 2D NMR
add	If TRUE, additional spectrum plots are overlaid with the current plot
showGrid	Shows a grid of data points. Defaults to FALSE
manualScale	If TRUE, scaling factor is taken from environment variables
plotTitle	Defines the main title of the plot
restrictToRange	Restrict plot area to range of available data points. Defaults to FALSE
currentSpectrumOriginal	Optional spectral data. If omitted, data from the environment variables is used
perspective	If TRUE, a perspective plot will be displayed for 2D data instead of the regular topographic view
noise	If provided, a line or plane at this level will be added to the plot to indicate noise level
dimension	"1D' or "2D". If not provided, this will be deduced from the data

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
        PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
        fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
        NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
```

plotPCA *A function for plotting PCA plots.*

Description

This function performs PCA, then plots PC1 and PC2.

Usage

```
plotPCA(
  mrbinResults,
  defineGroups = TRUE,
  loadings = FALSE,
  legendPosition = "bottomleft",
  annotate = TRUE,
  verbose = TRUE
)
```

Arguments

mrbinResults	An mrbin object
defineGroups	Should groups be colored differently?
loadings	Should loadings be plotted instead of scores?
legendPosition	Where should the legend be plotted, Defaults to "left", other options include "top", "topright", etc.
annotate	Should loadings be annotated with metabolite identities, if available in $metadata?
verbose	Should a summary be displayed?

Value

An invisible prcomp result object

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=FALSE,parameters=list(dimension="2D",
    binRegion=c(8,1,15,140),binwidth2D=0.1,binheight=4,solventRegion=c(5.5,4.2),
    PQNScaling="No",noiseRemoval="Yes",trimZeros="Yes",tryParallel=FALSE,
    fixNegatives="No",logTrafo="No",PCA="No",signal_to_noise2D=10,
    NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"),
                 system.file("extdata/2/12/pdata/10",package="mrbin"))))
plotPCA(mrbinResults)
```

plotResults *A function for plotting quality indicators, including PCA plots.*

Description

This function plots boxplots (bin-wise and sample-wise) as visual quality indicators. It also performs PCA, then plots PC1 and PC2 and loading plots.

Usage

```
plotResults(mrbinResults, defineGroups = TRUE, process = TRUE, silent = FALSE)
```

Arguments

mrbinResults	An mrbin object
defineGroups	Should group membership be highlighted in PCA?
process	If set to FALSE, the file name will be extended by "Raw" to indicate that data has not been processed yet
silent	If set to TRUE, plots will be saved but not shown for the binning step for speed purposes

Value

None

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=FALSE,parameters=list(dimension="2D",
    binRegion=c(8,1,15,140),binwidth2D=0.2,binheight=4,solventRegion=c(5.5,4.2),
    PQNScaling="No",noiseRemoval="Yes",trimZeros="Yes",tryParallel=FALSE,
    fixNegatives="No",logTrafo="No",PCA="No",signal_to_noise2D=10,
    NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"),
                 system.file("extdata/2/12/pdata/10",package="mrbin"))))
plotResults(mrbinResults)
```

PQNScaling *A function for PQN scaling.*

Description

This function performs PQN scaling. To further exclude unreliable noise, only the most intense signals are used. For 1H and 1H-13C HSQC spectra, most of the sugar regions can be excluded to avoid a dominating effect of the multiple glucose signals.

Usage

```
PQNScaling(
  NMRdata,
  ignoreGlucose = "Yes",
  dimension = "1D",
  ppmNames = "borders",
  sugarArea = c(5.4, 3.35, 72, 100),
  minimumFeatures = 40,
  showHist = FALSE,
  verbose = TRUE,
  errorsAsWarnings = FALSE
)
```

Arguments

NMRdata	A matrix containing NMR data or an mrbin object. Columns=frequencies,rows=samples
ignoreGlucose	A character value ("Yes" or "No")
dimension	A character value ("1D" or "2D")
ppmNames	A character value ("borders" or "mean")
sugarArea	A numeric vector defining the the borders of glucose area
minimumFeatures	A numeric value defining minimum feature number used
showHist	A logical value, default is FALSE
verbose	Should a summary be printed?
errorsAsWarnings	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible matrix or mrbin object containing scaled NMR data.

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
               binwidth1D=0.05,PQNScaling="No",PCA="No",tryParallel=FALSE,logTrafo="No",
                    NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                                system.file("extdata/2/10/pdata/10",package="mrbin"),
                                system.file("extdata/3/10/pdata/10",package="mrbin"))))
mrbinResults<-PQNScaling(mrbinResults)
```

predictWrapper *A function returning predicted values for use with the fia function.*

Description

This function predicts group membership and returns a numeric vector with results.

Usage

```
predictWrapper(
  model,
  dataSet,
  functionNamePredict = "predict",
  firstLevel = 1,
  parameterNameObject = "object",
  parameterNameData = "x",
  dataFrameFlag = FALSE,
  ...
)
```

Arguments

model	A predictive model. Make sure to have loaded all required packages before starting this function
dataSet	A matrix or dataframe containing data, depending on what your predict function requires. Columns=features, rows=samples
functionNamePredict	The name of the prediction function. This only needs to be changed if the prediction function is not called predict
firstLevel	Numeric value of first level or group. Usually 1 but for glm such as in the example this needs to be 0.
parameterNameObject	The name of the parameter for passing the model to the prediction function
parameterNameData	The name of the parameter for passing the data to the prediction function
dataFrameFlag	Logical value indicating whether the data object is a data frame rather than a matrix.
...	Optional, additional parameters that will be passed to the prediction function.

Value

A numeric (integer) vector of predicted group memberships.

Examples

```
#First, define group membership and create the example feature data
group<-factor(c(rep("Group1",4),rep("Group2",5)))
names(group)<-paste("Sample",1:9,sep="")
dataset<-data.frame(
  Feature1=c(5.1,5.0,6.0,2.9,4.8,4.6,4.9,3.8,5.1),
  Feature2=c(2.6,4.0,3.2,1.2,3.1,2.1,4.5,6.1,1.3),
  Feature3=c(3.1,6.1,5.8,5.1,3.8,6.1,3.4,4.0,4.4),
  Feature4=c(5.3,5.2,3.1,2.7,3.2,2.8,5.9,5.8,3.1),
  Feature5=c(3.2,4.4,4.8,4.9,6.0,3.6,6.1,3.9,3.5)
  )
rownames(dataset)<-names(group)
#train a model - here we use a logit model instead of ANN as a demonstration
mod<-glm(group~Feature1+Feature2+Feature3+Feature4+Feature5,
  data=data.frame(group=group,dataset),family="binomial")
predictWrapper(model=mod,dataSet=dataset,firstLevel=0,type="response")
```

---

printParameters *A function for printing parameters to the screen.*

Description

This function reads parameters from the global variable mrbin.env$mrbin$parameters and prints the required R code for creating a data set to the screen.

Usage

```
printParameters()
```

Value

None

Examples

```
printParameters()
```

---

putToEnv *A function for changing and adding variables in the package environment.*

Description

This function can change variables in the current package environment. This may be helpful for debugging or for some plotting functions.

Usage

```
putToEnv(variableList)
```

Arguments

variableList	A list containing all objects to be saved in the local package environment.

Value

None

Examples

```
putToEnv(list(bins=NULL))
```

---

readBruker *A function for reading Bruker NMR spectra.*

Description

This function reads Bruker NMR data. 1D and 2D data are supported.

Usage

```
readBruker(
  folder = NULL,
  dimension = NULL,
  onlyTitles = FALSE,
  useAsNames = "Spectrum titles",
  checkFiles = FALSE
)
```

Arguments

folder	Defines the exact NMR data folder. If NULL, mrbin parameter set is used
dimension	Defines the data dimension, "1D" or "2D". Only used if not NULL
onlyTitles	Read only spectrum titles, but no data. Defaults to FALSE
useAsNames	How should sample names be generated
checkFiles	Only check if the folder exists or contains NMR data. Defaults to FALSE

Value

An (invisible) list containing spectral data and the spectrum name

Examples

```
exampleData<-readBruker(folder=system.file("extdata/1/10/pdata/10",package="mrbin"),
                        dimension="1D")
```

recreatemrbin *A function recreating parameters from previous runs.*

Description

This function reads parameters from a text file that was created during a previous run or mrbin(). After reading, the data can be recreated using mrbin(). File names in $parameters might need to be updated.

Usage

```
recreatemrbin(filename = NULL)
```

Arguments

filename	File path/name of the mrbin parameter file to be loaded

Value

None

Examples

```
# Insert full folder path and file name
recreatemrbin(system.file("extdata/mrbin.txt",package="mrbin"))
```

removeFromPlot *A function for removing NMR spectra from the plot list.*

Description

This function removes a spectrum from the plot list.

Usage

```
removeFromPlot(folder = NULL, dimension = "1D")
```

Arguments

folder	Defines the exact NMR data folder.
dimension	Defines the data dimension, "1D" or "2D".

Value

none

Examples

```
removeFromPlot()
```

`removeNoise` *A function for removing bins below noise level.*

Description

This function checks for each bin (column) whether its level is below the individual noise level times the signal-to-noise ratio. If less than the defined threshold level are above noise*SNR, the whole bin is removed.

Usage

```
removeNoise(mrbinResults, verbose = TRUE, errorsAsWarnings = FALSE)
```

Arguments

`mrbinResults`	An mrbin object
`verbose`	Should a summary be printed?
`errorsAsWarnings`	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible mrbin object

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
                    binwidth1D=0.05,noiseRemoval="No",PQNScaling="No",tryParallel=FALSE,
                    fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
                    NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                                 system.file("extdata/2/10/pdata/10",package="mrbin"),
                                 system.file("extdata/3/10/pdata/10",package="mrbin"))))
mrbinResults<-removeNoise(mrbinResults)
```

removeSpectrum *A function for removing a spectrum.*

Description

This function lets the user pick spectra from a list for removal from data analysis.

Usage

```
removeSpectrum(
  mrbinResults = NULL,
  spectra = NULL,
  verbose = TRUE,
  errorsAsWarnings = FALSE
)
```

Arguments

mrbinResults	An mrbin object. If not provided, the function works on the package environment
spectra	Character vector with NMR folder names to be excluded. If provided, no interactive selection will be shown
verbose	Should a summary be printed?
errorsAsWarnings	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible mrbin object (only if an mrbin object was provided)

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
         binwidth1D=0.05,PQNScaling="No",PCA="No",tryParallel=FALSE,logTrafo="No",
         noiseRemoval="No",
         NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                      system.file("extdata/2/10/pdata/10",package="mrbin"),
                      system.file("extdata/3/10/pdata/10",package="mrbin"))))
mrbinResults<-removeSpectrum(mrbinResults,
 spectra=c(system.file("extdata/2/10/pdata/10",package="mrbin")))
```

resetEnv *A parameter resetting function*

Description

This function resets the parameter variables.

Usage

```
resetEnv()
```

Value

None

Examples

```
resetEnv()
```

right *A function for changing plotNMR plots.*

Description

This function moves right the plot region of the current NMR plot.

Usage

```
right(refreshPlot = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
mrbin(silent=TRUE,parameters=list(dimension="1D",binwidth1D=.5,
         noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         PQNScaling="No",saveFiles="No",referenceScaling="No",
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
zoomIn()
right()
```

setCurrentSpectrum *A function for interactively setting the current spectrum.*

Description

This function lets the user pick a spectrum from the list of spectra analysis. This function is meant only for use within the mrbin function.

Usage

```
setCurrentSpectrum(spectrumNumber = NULL)
```

Arguments

spectrumNumber	If provided, this number will be used; defaults to NULL

Value

None

Examples

```
setCurrentSpectrum(spectrumNumber=1)
```

setDilutionFactors *A function for setting dilution factors.*

Description

This function edits the dilution factors of an mrbin object but does not change the bin data.

Usage

```
setDilutionFactors(
  mrbinObject,
  dilutionFactors = NULL,
  errorsAsWarnings = FALSE,
  alwaysShowOptionKeep = FALSE
)
```

Arguments

mrbinObject	An mrbin object
dilutionFactors	An optional vector of dilution factors. If provided, no user input is requested
errorsAsWarnings	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.
alwaysShowOptionKeep	If TRUE, you will be asked to keep current values even if they do not match the current dataset.

Value

An invisible mrbin object

Examples

```
mrbinObject<-mrbin(silent=TRUE,
                   parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
               binwidth1D=0.04,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,0.5,10,156),
                   saveFiles="No",referenceScaling="No",noiseRemoval="No",
                   fixNegatives="No",logTrafo="No",noiseThreshold=.05,tryParallel=FALSE,
                   NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                              system.file("extdata/3/10/pdata/10",package="mrbin"))
                   ))
mrbinObject<-setDilutionFactors(mrbinObject,dilutionFactors=c(1.5,2))
```

---

setNoiseLevels *A function for setting and plotting noise levels.*

Description

This function reads parameters from the global variable mrbin.env$mrbin$parameters and plots exemplary spectra and respective noise levels. Plots will be saved if saveFiles is set to "Yes".

Usage

```
setNoiseLevels(
  mrbinObject,
  plotOnly = FALSE,
  showSpectrumPreview = NULL,
  silent = FALSE
)
```

Arguments

mrbinObject	An mrbin object
plotOnly	Should only noise plots be generated (TRUE), or should noise levels be adjusted interactively (FALSE)
showSpectrumPreview	Should plots be shown? If not provided, this value will be taken from the mrbin object parameters
silent	If set to TRUE, plots will not be shown but might still be saved

Value

An invisible mrbin object

Examples

```
mrbinObject<-mrbin(silent=TRUE,
                   parameters=list(verbose=TRUE,dimension="1D",PQNScaling="No",
              binwidth1D=0.04,signal_to_noise1D=1,PCA="No",binRegion=c(9.5,0.5,10,156),
                   saveFiles="No",referenceScaling="No",noiseRemoval="No",
                   fixNegatives="No",logTrafo="No",noiseThreshold=.05,tryParallel=FALSE,
                   NMRfolders=c(system.file("extdata/2/10/pdata/10",package="mrbin"),
                              system.file("extdata/3/10/pdata/10",package="mrbin"))
                   ))
mrbinObject<-setNoiseLevels(mrbinObject,plotOnly=TRUE)
```

---

setOffset *A function for changing plotNMR plots.*

Description

This function moves up or down the 1D plot region of the current NMR plot.

Usage

```
setOffset(offsetValue = NULL)
```

Arguments

offsetValue	The new offset value. Defaults to NULL

Value

None

Examples

```
setOffset(0)
```

setParam *A function setting parameters and checking for consistency.*

Description

This function set parameters and checks parameters for consistency.

Usage

```
setParam(parameters = NULL, metadata = NULL)
```

Arguments

parameters	List of parameters to be set
metadata	List of metadata to be set

Value

None

Examples

```
setParam(parameters=list(dimension="1D"))
```

timeStampMrbin *A function for time stamping mrbin objects.*

Description

This function adds time stamps to an mrbin object and returns it. Is used only within functions making changes to mrbin objects.

Usage

```
timeStampMrbin(
  mrbinObject,
  functionName = "InProgress...",
  versionNumber = "0",
  changeDetails = "InProgress...",
  steps = 0,
  comment = ""
)
```

Arguments

mrbinObject	An mrbin object
functionName	Name of the package and function calling this command
versionNumber	Version number of the package calling this command
changeDetails	Details of changes made to the mrbin object
steps	Indicates which step to perform: 0 (only pre-change), 1 (only post-change)
comment	An optional character vector describing the change

Value

An (invisible) mrbin object

Examples

```
mrbinObject<-createmrbin()
mrbinObject<-timeStampMrbin(mrbinObject)
```

---

trimZeros *A function for trimming zero-values bins.*

Description

This function removes zero-values bins. These might be created during removal of solvent and additional areas, or at the edges of the spectrum.

Usage

```
trimZeros(mrbinResults)
```

Arguments

mrbinResults	An mrbin object

Value

An invisible mrbin object

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D", logTrafo="No",
                    binwidth1D=0.05,signal_to_noise1D=50, verbose=TRUE, PCA="No",
                      trimZeros="No",tryParallel=FALSE,
                    NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                                 system.file("extdata/2/10/pdata/10",package="mrbin"))))
mrbinResults<-trimZeros(mrbinResults)
```

unitVarianceScaling *A function for scaling to unit variance.*

Description

This function performs scaling of binned data to unit variance so that each bin has variance 1 and mean 0. This is rarely necessary, but might be advantageous, e.g. in artificial neural networks.

Usage

```
unitVarianceScaling(mrbinResults, verbose = TRUE, errorsAsWarnings = FALSE)
```

Arguments

mrbinResults	An mrbin object
verbose	Should a summary be printed?
errorsAsWarnings	If TRUE, errors will be turned into warnings. Should be used with care, as errors indicate undocumented changes to the data.

Value

An invisible mrbin object containing scaled NMR data.

Examples

```
mrbinResults<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
         binwidth1D=0.05,PQNScaling="No",PCA="No",tryParallel=FALSE,logTrafo="No",
         NMRfolders=c(system.file("extdata/1/10/pdata/10",package="mrbin"),
                      system.file("extdata/2/10/pdata/10",package="mrbin"),
                      system.file("extdata/3/10/pdata/10",package="mrbin"))))
mrbinResults<-unitVarianceScaling(mrbinResults)
```

up *A function for changing plotNMR plots.*

Description

This function moves up the plot region of the current NMR plot (only 2D).

Usage

```
up(refreshPlot = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
resetEnv()
mrbin(silent=TRUE,parameters=list(dimension="2D",binwidth2D=0.5,
         binheight=3,PQNScaling="No",referenceScaling="No",binRegion=c(4,3,60,65),
         noiseRemoval="No",trimZeros="No",cropHSQC="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
         NMRfolders=c(system.file("extdata/1/12/pdata/10",package="mrbin"))))
plotNMR()
zoomIn()
up()
```

---

zoom *A function for changing plotNMR plots.*

Description

This function changes the plot region of the current NMR plot. Can be called with no arguments: zoom(). In this case the user will be asked for manual input.

Usage

```
zoom(left = NULL, right = NULL, top = NULL, bottom = NULL, refreshPlot = TRUE)
```

Arguments

left	New left boundary
right	New right boundary
top	New top boundary
bottom	New bottom boundary
refreshPlot	Refresh plot automatically. Defaults to TRUE

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
         PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
zoom(left=4.6,right=2,top=10,bottom=150)
```

zoomIn *A function for changing plotNMR plots.*

Description

This function zooms into the plot region of the current NMR plot.

Usage

```
zoomIn(refreshPlot = TRUE, x = TRUE, y = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE
x	Change x axis? Defaults to TRUE
y	Change y axis? Defaults to TRUE

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
        PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
        fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
        NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
zoomIn()
```

zoomOut *A function for changing plotNMR plots.*

Description

This function zooms out from the plot region of the current NMR plot.

Usage

```
zoomOut(refreshPlot = TRUE, x = TRUE, y = TRUE)
```

Arguments

refreshPlot	Refresh plot automatically. Defaults to TRUE
x	Change x axis? Defaults to TRUE
y	Change y axis? Defaults to TRUE

Value

None

Examples

```
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
         PQNScaling="No",noiseRemoval="No",trimZeros="No",tryParallel=FALSE,
         fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
         NMRfolders=system.file("extdata/1/10/pdata/10",package="mrbin")))
plotNMR()
zoomIn()
zoomOut()
```

Index