

# Package ‘mrbin’

May 9, 2026

**Title** Metabolomics Data Analysis Functions

**Version** 1.9.5

**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](https://doi.org/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, Matrix

**Depends** R (>= 2.10)

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Suggests** parallel, keras

**VignetteBuilder** utils

**URL** <https://github.com/kleinomicslab/mrbin>

**NeedsCompilation** no

**Author** Matthias Klein [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-7455-5381>>)

**Maintainer** Matthias Klein <[matthias.s.klein@gmx.net](mailto:matthias.s.klein@gmx.net)>

**Repository** CRAN

**Date/Publication** 2026-02-27 14:20:07 UTC

## Contents

addToPlot	3
annotatemrbin	4
atnv	6
checkmrbin	7
contMin	8
contPlus	8
createAnnotationList	9
createmrbin	10
cropNMR	10
dilutionCorrection	11
down	12
editmetabolitesmrbin	12
editmrbin	13
fia	15
getEnv	17
intMin	17
intPlus	18
left	19
logTrafo	19
metadatamrbin	20
mrbin	21
mrbinrun	22
mrheatmap	23
mrplot	25
plotMultiNMR	27
plotNMR	29
plotPCA	31
plotResults	32
PQNScaling	33
predictWrapper	35
printParameters	36
putToEnv	37
readBruker	37
recreatemrbin	38
removeFromPlot	39
removeNoise	39
removeSpectrum	40
resetEnv	41
right	41
setCurrentSpectrum	42
setDilutionFactors	43
setNoiseLevels	44
setOffset	45
setParam	45
timeStampMrbin	46
trimZeros	47

<code>addToPlot</code>	3
<code>unitVarianceScaling</code> . . . . .	47
<code>up</code> . . . . .	48
<code>zoom</code> . . . . .	49
<code>zoomIn</code> . . . . .	50
<code>zoomOut</code> . . . . .	51
<b>Index</b>	<b>52</b>

---

<code>addToPlot</code>	<i>A function for adding NMR spectra to the plot list.</i>
------------------------	--

---

## Description

This function adds a spectrum to the plot list.

## Usage

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

## Arguments

<code>folder</code>	Defines the exact NMR data folder. If NULL, <code>mrbin</code> parameter set is used
<code>dimension</code>	Defines the data dimension, "1D" or "2D". Only used if not NULL
<code>NMRvendor</code>	Defines the NMR manufacturer, default is "Bruker"
<code>useAsNames</code>	How should sample names be generated
<code>add</code>	Add spectra to existing list, or replace existing spectra. Default is TRUE
<code>omitCurrent</code>	Omit the "current spectrum" spot and start filling the additional lists immediately. Default is FALSE

## 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,
  annotate = TRUE,
  binwidth = 0.01,
  binheight = 1,
  dimension = "1D",
  metaboliteIdentities = NULL,
  add = FALSE,
  hideChemicalShift = FALSE,
  includeAll = FALSE,
  maxMetabolites = NULL,
  batchSize = 50,
  tiers = c(1, 2, 4, 5),
  peakRatio = c(0.67, 0.4, 0),
  Rmodifier = NULL,
  confirmationRthreshold = c(0.89, 0.84, 0.82),
  peakRthreshold = NULL,
  showMissing = FALSE,
  exclude = NULL,
  expandRange = 0,
  expandRangeFactor = 15,
  verbose = TRUE
)

```

### Arguments

mrbinObject	An mrbin object or matrix of bin intensities. In case of a matrix, the row names can be in the format left border, right border ("1.22,1.21") or the center of the bin ("1.215"). In the latter case, the parameters binwidth (and binheight if required) should be chosen accordingly. For 2D data, this would be "1.22,1.21,35,36" or "1.215,35.5"
annotate	If FALSE, the mrbin object will not be changed.
binwidth	Full width of each bin. Will only be used if mrbinObject is a matrix
binheight	Full height of each bin. Will only be used if mrbinObject is a matrix and dimension is set to "2D"
dimension	Dimension of NMR data set, option are "1D" or "2D" (e.g. for HSQC data). Will only be used if mrbinObject is a matrix

metaboliteIdentities	A numeric matrix or the file path for a .csv file containing such a matrix, the first columns containing metabolite names and the first row being a header. Each row belongs to one unique metabolite signal (left, right, top, bottom borders). Row names are metabolite names. If provided, this will overwrite any current metaboliteIdentities matrix present in the mrbin object. If missing, data currently attached to the mrbin object (if any) will be used.
add	Should the new metabolite list be added to an existing list, or replace the current list?
hideChemicalShift	Should the chemical shift (bin borders) of an identified metabolite be removed, leaving only the metabolite id, or should both be shown? Showing both helps in identifying signals of interest, but hiding the chemical shift might make better plots.
includeAll	Should all metabolites in the predefined list be used, even if they aren't in the ordered list for this sample matrix?
maxMetabolites	Maximum number of metabolites to be used. Default is NULL (all metabolites in the list will be used)
batchSize	Look for this number of top metabolites first, then move on the next batch of lower-abundance metabolites
tiers	Should all tiers (1 through 5) of identification be applied? .
peakRatio	A threshold for the minimum ratio of significant peaks to total peaks to confirm an annotation
Rmodifier	A number by which the confirmationRthreshold is multiplied. This can be used to lower R threshold when signal-to-noise levels are low, such as in 2D spectra. If set to NULL (default), this will use 1 for 1D spectra and 0.4 for 2D spectra
confirmationRthreshold	A threshold to define the r-value cutoff to confirm an annotation
peakRthreshold	A threshold to define the r-value cutoff that bins within one peak need to show to the first identified bin of the peak to be considered. If NULL, this will use .8 for 1D and .2 for 2D
showMissing	Should metabolites missing from the peak database be shown
exclude	Exclude specific metabolites from annotation, e.g. "DSS" if TSP was used instead.
expandRange	Expand the search range by this value in ppm. This could be used for low-field spectra.
expandRangeFactor	The expandRange will be multiplied by this value for the second, indirect dimension, e.g 13C.
verbose	Should outputs be shown or suppressed?

**Value**

An (invisible) mrbin object

## Examples

```

resetEnv()#clean up previous data from the package environment
results<-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,
    NMRvendor="mrbin",
    example=TRUE,#only used for the package examples
    NMRfolders=c(system.file("extdata/3.mr1",package="mrbin"),
      system.file("extdata/2.mr1",package="mrbin"),
      system.file("extdata/1.mr1",package="mrbin")))
metaboliteIdentities=matrix(c(1.346,1.324,21,23,1,1,
  3.052,3.043,30.5,33.5,1,1),
  ncol=6,byrow=TRUE)
rownames(metaboliteIdentities)=c("Lactate","Creatinine")
colnames(metaboliteIdentities)=c("left","right","top","bottom","usePeak1D","usePeak2D")
results<-annotatemrbin(results,metaboliteIdentities=metaboliteIdentities)
results$metadata$annotations[150:160]
plotPCA(results,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  $0.01 \times$  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 use 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()#clean up previous data from the package environment
results<-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,NMRvendor="mrbin",
                               example=TRUE,#only used for the package examples
                               NMRfolders=c(system.file("extdata/2.mr1",package="mrbin"),
                                             system.file("extdata/3.mr1",package="mrbin"))
               ))
sum(results$bins<=0)
exampleNMRpositive<-atnv(NMRdata=results$bins, noiseLevels=results$parameters$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	<i>A function for changing plotNMR plots.</i>
---------	---

---

**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()#clean up previous data from the package environment
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",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr2",package="mrbin"),
  NMRvendor="mrbin"))
plotNMR()
contMin()
```

---

contPlus	<i>A function for changing plotNMR plots.</i>
----------	---

---

**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()#clean up previous data from the package environment
addToPlot(folder=system.file("extdata/1.mr2",package="mrbin"),dimension="2D",NMRvendor="mrbin")
plotNMR()
contPlus()
```

---

createAnnotationList *A function for creating annotation lists.*

---

**Description**

This function creates a matrix of peak positions for use with the `annotatemrbin` function

**Usage**

```
createAnnotationList(
  metaboliteNames,
  binwidth = 0.01,
  binheight = 1,
  verbose = TRUE
)
```

**Arguments**

metaboliteNames	Vector of metabolite names
binwidth	Full width of each bin. Will only be used if <code>mrbinObject</code> is a matrix
binheight	Full height of each bin. Will only be used if <code>mrbinObject</code> is a matrix and dimension is set to "2D"
verbose	Should outputs be shown or suppressed?

**Value**

A numeric matrix, the first columns containing metabolite names and the first row being a header. Each row belongs to one unique metabolite signal (left, right, top, bottom borders). Row names are metabolite names.

**Examples**

```
resetEnv()#clean up previous data from the package environment
```

---

createmrbin	<i>A function for creating mrbin objects.</i>
-------------	---

---

**Description**

This function creates an mrbin object and returns it.

**Usage**

```
createmrbin()
```

**Value**

An (invisible) mrbin object

**Examples**

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

---

cropNMR	<i>A function for cropping HSQC spectra.</i>
---------	--

---

**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()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,
  parameters=list(dimension="2D",binwidth2D=1,binheight=4,cropHSQC="No",PCA="No",
  PQNScaling="No",noiseRemoval="No",removeSolvent="No",verbose=TRUE,
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr2",package="mrbin"),NMRvendor="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

```
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
  binwidth1D=0.05,PQNScaling="No",PCA="No",logTrafo="No",
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",package="mrbin")),
  metadata=list(dilutionFactors=c(.75,1,.5)))
results<-dilutionCorrection(results)
```

---

down	<i>A function for changing plotNMR plots.</i>
------	---

---

**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()#clean up previous data from the package environment
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",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr2",package="mrbin"),
  NMRvendor="mrbin"))

plotNMR()
zoomIn()
down()
```

---

editmetabolitesmrbin	<i>A function for editing metabolite identities.</i>
----------------------	--

---

**Description**

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

**Usage**

```
editmetabolitesmrbin(
  mrbinObject,
  borders = NULL,
  metabolitenames = NULL,
  add = FALSE,
  ids = NULL
)
```

**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?
ids	A matrix of potential metabolite identities. This has to be a matrix with columns indicating left, right, top, bottom. Rownames are metabolite names. If this matrix is not provided, borders and metabolitenames have to be provided.

**Value**

An (invisible) mrbin object

**Examples**

```

resetEnv()#clean up previous data from the package environment
results<-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,
    NMRvendor="mrbin",
    example=TRUE,#only used for the package examples
    NMRfolders=c(system.file("extdata/2.mr1",package="mrbin"),
      system.file("extdata/3.mr1",package="mrbin"))
  ))
results<-editmetabolitesmrbin(results,borders=matrix(c(
  1.346,1.324,
  4.12,4.1,
  3.052,3.043,
  4.066,4.059
),ncol=2,byrow=TRUE),metabolitenames=c(
  "Lactate",
  "Lactate",
  "Creatinine",
  "Creatinine"
))
results$parameters$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

<code>model</code>	A predictive model. Make sure to have loaded all required packages before starting this function
<code>dataSet</code>	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
<code>factors</code>	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
<code>nSeed</code>	Number of times that the test will be repeated, selecting different random features
<code>numberOfSamples</code>	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")
fiareults<-fia(model=mod,dataSet=dataset,factors=group,parameterNameData="newdata",
  firstLevel=0,type="response")
fiareults$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**

```

resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",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**

```

resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",package="mrbin")))
plotNMR()
intPlus()

```

---

left	<i>A function for changing plotNMR plots.</i>
------	---

---

**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()#clean up previous data from the package environment
mrbin(silent=TRUE,parameters=list(dimension="1D",binwidth1D=.5,
  noiseRemoval="No",trimZeros="No",
  PQNScaling="No",saveFiles="No",referenceScaling="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",package="mrbin")))
plotNMR()
zoomIn()
left()
```

---

logTrafo	<i>A function for log transforming data.</i>
----------	--

---

**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()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D", logTrafo="No",
  binwidth1D=0.05,signal_to_noise1D=50,verbose=TRUE,PCA="No",NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"))))
results<-logTrafo(results)
```

---

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, graphics = graphics)
```

**Arguments**

mrbinResults	An mrbin object
metadata	An optional list of objects to be changed. If provided, interactive mode is deactivated
graphics	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**Value**

An invisible mrbin object

## Examples

```

resetEnv()#clean up previous data from the package environment
results<-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,NMRvendor="mrbin",
    example=TRUE,#only used for the package examples
    NMRfolders=c(system.file("extdata/2.mr1",package="mrbin"),
      system.file("extdata/3.mr1",package="mrbin"))
  ))
results<-metadatamrbin(results,metadata=list(projectTitle="Test project"))

```

---

mrbin	<i>A function setting the parameters and performing binning and data processing</i>
-------	---

---

## 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,
  graphics = TRUE
)

```

## 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.
graphics	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**Value**

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

**Examples**

```
# Set parameters in command line.
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(
  dimension="1D",binwidth1D=0.01,
  signal_to_noise1D=25,noiseThreshold=0.75,useAsNames="Spectrum titles",
  example=TRUE,#only used for the package examples
  NMRvendor="mrbin",NMRfolders=c(
    system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",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,
  graphics = TRUE
)
```

**Arguments**

<code>createbins</code>	If TRUE, new bin data is generated
<code>process</code>	If TRUE, bin data is processed, e.g. by noise removal, atnv, etc.
<code>mrbinResults</code>	An mrbin object. Needs to be provided only if <code>createbins</code> is FALSE
<code>silent</code>	If set to FALSE, no new time calculation is performed
<code>graphics</code>	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**Value**

An invisible mrbin object

**Examples**

```
resetEnv()#clean up previous data from the package environment
setParam(parameters=list(dimension="2D",binwidth2D=0.1,binheight=5,
  binRegion=c(8,1,15,140),PQNScaling="No",useAsNames="Spectrum titles",
  fixNegatives="No",logTrafo="No",signal_to_noise2D=10,solventRegion=c(5.5,4.2),
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr2",package="mrbin"),
    system.file("extdata/2.mr2",package="mrbin"),
    system.file("extdata/3.mr2",package="mrbin")),
  NMRvendor="mrbin"))
results<-mrbinrun()
```

---

 mrheatmap

*A function for plotting heatmaps.*


---

**Description**

This function plots heatmaps based on rank order, using heatmap from the stats package

**Usage**

```
mrheatmap(
  results,
  binlist = NULL,
  samplelist = NULL,
  annotate = FALSE,
  cexRow = 0.7,
  cexCol = 0.8,
  margins = c(4, 6),
  Colv = NA,
  Rowv = NULL,
  closeDevice = TRUE,
  ...
)
```

**Arguments**

results	Either an mrbin object or a numeric matrix containing sample names as rownames and feature names as column names.
binlist	A vector containing bin names as they are generated by mrbin (colnames). If provided, only these columns will be shown.
samplelist	A vector containing sample names (rownames). If provided, only these rows will be shown.

annotate	Should peak annotations be shown? This requires annotation data in the mrbin object.
cexRow	Font size for row labels
cexCol	Font size for column labels
margins	Determines the plot margins.
Colv	Determines if and how the column dendrogram should be computed and reordered. Default: NA (dendrogram will not be used)
Rowv	Determines if and how the row dendrogram should be computed and reordered. Default: NULL (dendrogram will be used)
closeDevice	Should previous plots be closed prior to plotting?
...	Additional graphical parameters that will be passed to the stats function heatmap

### Value

None

### Examples

```

resetEnv()#clean up previous data from the package environment
# First create NMR bin data, then plot some differential bins.
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(verbose=FALSE,
  dimension="1D",binwidth1D=0.01,PCA="No",showSpectrumPreview="No",
  signal_to_noise1D=25,noiseThreshold=0.75,useAsNames="Spectrum titles",
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
  system.file("extdata/2.mr1",package="mrbin"),
  system.file("extdata/3.mr1",package="mrbin"))
))
metadata<-c(0,0,1)
#Find significant signals
pvalues<-rep(NA,ncol(results$bins))
names(pvalues)<-colnames(results$bins)
for(i in 1:ncol(results$bins)){
  model<-stats::lm(intensity~treatment,
    data=data.frame(intensity=results$bins[,i],treatment=metadata))
  pvalues[i]<-stats::anova(model)$"Pr(>F)"[1]
}
significantBins<-names(sort(pvalues)[1:30])
metaboliteIdentities=matrix(c(1.346,1.324,21,23,
  3.052,3.043,30.5,33.5,
  5.7,6.0,0,150),
  ncol=4,byrow=TRUE)
#Annotate the dataset with signal identities
rownames(metaboliteIdentities)=c("Lactate","Creatinine","Urea")
results<-annotatemrbin(results,metaboliteIdentities=metaboliteIdentities)
mrheatmap(results=results,
  binlist=significantBins,annotate=TRUE,
  main="Significant signals")

```

---

`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,  
  NMRvendor = "Bruker",  
  dimensions = "1D",  
  intensity1D = NULL,  
  zoom = NULL,  
  color = NULL,  
  background = NULL,  
  lwd = 1,  
  plotTitle = "",  
  showNames = "Spectrum titles",  
  graphics = TRUE,  
  highlight = NULL,  
  binlist = NULL,  
  annotate = NULL,  
  metaboliteIdentities = NULL,  
  annotateColors = c("black", "red", "orange3", "yellow4", "green3", "blue", "purple",  
    "violet", "brown4", "chartreuse4", "blue4", "deeppink", "orangered", "olivedrab",  
    "cadetblue", "tomato3"),  
  annotateAngles = c(35, -35, 20, -20, 45, -45, 60, -60, 75, -75, 15, -15),  
  hideExcludedAnnotations = FALSE,  
  ...  
)
```

### Arguments

<code>hideMenu</code>	Do not show the menu. Defaults to FALSE
<code>folders</code>	Optional vector of folder names of spectra to load. Defaults to NULL
<code>NMRvendor</code>	Optional value naming the format of spectra to load. Defaults to Bruker
<code>dimensions</code>	Optional vector dimensions of spectra to load. Defaults to NULL
<code>intensity1D</code>	Optional value of initial 1D intensity. Defaults to NULL
<code>zoom</code>	Optional vector of initial zoom area. Defaults to NULL
<code>color</code>	Defines the color of the spectrum plot. If NULL, a rainbow theme is used for 2D NMR

background	Background color, defaults to NULL (no background fill, usually results in a white background)
lwd	Line width, defaults to 1
plotTitle	Plot title, defaults to "" (empty)
showNames	Display list of spectrum titles in plot, defaults to "Spectrum titles". Other options are "" and "Folder names"
graphics	Controls whether pop-up windows are shown for selections. Defaults to TRUE.
highlight	A vector of up to 2 frequencies that will be highlighted in the plot. If 2 values are provided the distance in Hz is shown as well. Defaults to NULL.
binlist	Optional: A vector containing bin names as they are generated by mrbin. These bins will be marked by rectangles in the plot. This could be useful for metabolite identification when having a list of significantly changing signals. Default is NULL.
annotate	Should peak annotation regions be shown?
metaboliteIdentities	Optional: A file path or 4-column matrix where each row belongs to one unique metabolite signal (left, right, top, bottom borders). Row names are metabolite names. For a file, this needs to be the file path for a .csv file containing such a matrix, the first columns containing metabolite names and the first row being a header. Each row belongs to one unique metabolite signal (left, right, top, bottom borders). Row names are metabolite names.
annotateColors	Colors for annotation boxes
annotateAngles	Angles for shading of annotation boxes
hideExcludedAnnotations	Should excluded peak annotation regions be hidden?
...	Additional graphical parameters that will be passed to the functions plot, lines, and/or contour

**Value**

None

**Examples**

```

resetEnv()#clean up previous data from the package environment
metaboliteIdentities=matrix(c(1.346,1.324,21,23,1,1,
                             4.12,4.1,70.8578,71.653,0,1,
                             3.052,3.043,30.5,33.5,1,1,
                             4.066,4.059,57,59.5,1,0,
                             2.582,2.479,46,49,1,1,
                             2.737,2.634,46,49,1,1),
                             ncol=6,byrow=TRUE)
rownames(metaboliteIdentities)=c("Lactate","Lactate","Creatinine","Creatinine","Citrate","Citrate")
colnames(metaboliteIdentities)=c("left","right","top","bottom","usePeak1D","usePeak2D")
mrplot(folders=c(system.file("extdata/1.mr2",package="mrbin"),
                        system.file("extdata/1.mr1",package="mrbin")),

```

```

        system.file("extdata/2.mr1",package="mrbin"),
        system.file("extdata/3.mr1",package="mrbin")),
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  dimensions=c("2D","1D","1D","1D"),zoom=c(2.8,2.4,20,60),
  highlight=c(2.564,2.537),
  binlist=c("2.725,2.675","2.575,2.525"),
  annotate=TRUE,metaboliteIdentities=metaboliteIdentities,
  plotTitle="Significant Bins",intensity1D=24,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("darkseagreen3", "orange", "blue", "red", "yellow", "gray",
    "purple"),
  rectangleColors2D = NULL,
  density = NULL,
  angles = 35,
  cex.axis = 0.7,
  rectangleFront = FALSE,
  correctOffset2D = TRUE,
  polygonRegion = NULL,
  maxPlots = Inf,
  setContours = TRUE,
  color = NULL,
  add = FALSE,
  showGrid = FALSE,
  buffer = TRUE,
  manualScale = TRUE,
  plotTitle = "",
  renewSpectrum = TRUE,
  restrictToRange = FALSE,
  enableSplit = TRUE,
  dimension = NULL,
  lwd = 1,
  background = NULL,

```

```

    titles = NULL,
    plotCurrent = TRUE,
    ...
)

```

### Arguments

region	A vector defining the plot region (left, right, top, bottom) or "all" for the whole spectrum
rectangleRegions	A 4-column matrix defining areas where to plot rectangles
rectangleColors	Define colors for the rectangles
rectangleColors2D	Define colors for rectangles in 2D spectra. If NULL, defaults to the same as rectangleColors
density	Shading lines for the rectangles
angles	Angles of shading lines for the rectangles
cex.axis	Font size of axis tick labels.
rectangleFront	Plot rectangles in front of spectrum rather than in background (only 2D)
correctOffset2D	Do a basic offset correction so 2D spectra have a baseline close to 0. Defaults to TRUE
polygonRegion	Defines 4 corners of a polygon to be plotted
maxPlots	The maximum number of 2D plots to be overlaid
setContours	Should upper and lower contour levels be calculated of the old ones be reused? Default: TRUE
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
buffer	Speed up plotting by loading a plot. Defaults to TRUE
manualScale	If TRUE, scaling factor is taken from environment variables
plotTitle	Defines the main title of the plot
renewSpectrum	Should a new size-reduced spectrum for quicker plotting be calculated, or can the old one be used? Default: TRUE
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
lwd	Line width, defaults to 1
background	Background color, defaults to NULL (no background fill, usually results in a white background)

titles	Display list of spectrum titles in plot, defaults to NULL
plotCurrent	Should the first (current) spectrum in the list be plotted, defaults to TRUE
...	Additional graphical parameters that will be passed to the functions plot, lines, and/or contour

**Value**

None

**Examples**

```

resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,NMRvender="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",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("darkseagreen3", "orange", "blue", "red", "yellow", "gray",
    "purple"),
  rectangleColors2D = NULL,
  density = NULL,
  angles = 35,
  rectangleFront = FALSE,
  polygonRegion = NULL,
  color = NULL,
  add = FALSE,
  showGrid = FALSE,
  manualScale = TRUE,
  plotTitle = "",
  title = NULL,

```

```

titleCounter = NULL,
hideNegative = FALSE,
restrictToRange = FALSE,
currentSpectrumOriginal = NULL,
spectrumTMP = NULL,
renewSpectrum = TRUE,
cex.axis = 0.7,
perspective = FALSE,
noise = NULL,
dimension = NULL,
plotDelay = 0.1,
lwd = 1,
background = NULL,
...
)

```

### Arguments

region	A vector defining the plot region (left, right, top, bottom) or "all" for the whole spectrum
rectangleRegions	A 4-column matrix defining areas where to plot rectangles
rectangleColors	Define colors for the rectangles
rectangleColors2D	Define colors for rectangles in 2D spectra. If NULL, defaults to the same as rectangleColors
density	Shading lines for the rectangles
angles	Angles of shading lines 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
title	Display the spectrum title in plot, defaults to NULL
titleCounter	Count of the spectrum title for positioning in plot, defaults to NULL
hideNegative	Should negative parts of the 2D spectrum be hidden? Defaults to FALSE
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

spectrumTMP	A size-reduced spectrum for quicker plotting. Defaults to NULL
renewSpectrum	Should a new size-reduced spectrum for quicker plotting be calculated, or can the old one be used? Default: TRUE
cex.axis	Font size of axis tick labels.
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
plotDelay	Add a small delay in seconds to force RStudio to update plots
lwd	Line width, defaults to 1
background	Background color, defaults to NULL (no background fill, usually results in a white background)
...	Additional graphical parameters that will be passed to the functions plot, lines, and/or contour

**Value**

An (invisible) dimension-reduced spectrum, either a matrix or a vector

**Examples**

```
resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",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,
    xpd = NA,
    cex = 1
  )

```

### 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 \$meta-data?
verbose	Should a summary be displayed?
xpd	Should labels be clipped to the plot region (TRUE) or exceed to margins (NA)
cex	Adjust text in plots by this factor

### Value

An invisible prcomp result object

### Examples

```

resetEnv()#clean up previous data from the package environment
results<-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",
  fixNegatives="No",logTrafo="No",PCA="No",signal_to_noise2D=10,
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr2",package="mrbin"),
    system.file("extdata/2.mr2",package="mrbin"),
    system.file("extdata/3.mr2",package="mrbin")),
  NMRvendor="mrbin"))
plotPCA(results)

```

---

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,  
  maxFeatures = 600  
)
```

**Arguments**

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

**Value**

None

**Examples**

```
resetEnv()#clean up previous data from the package environment  
results<-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",  
  fixNegatives="No",logTrafo="No",PCA="No",signal_to_noise2D=10,  
  example=TRUE,#only used for the package examples  
  NMRfolders=c(system.file("extdata/1.mr2",package="mrbin"),  
    system.file("extdata/2.mr2",package="mrbin")),  
  NMRvendor="mrbin"))  
plotResults(results)
```

---

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**

```
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
  binwidth1D=0.05,PQNScaling="No",PCA="No",logTrafo="No",
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",package="mrbin"))))
results<-PQNScaling(results)
```

---

predictWrapper	<i>A function returning predicted values for use with the fia function.</i>
----------------	---

---

### 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(verbose = TRUE)
```

**Arguments**

`verbose`            Should the code be shown on the screen?

**Value**

An invisible character string

**Examples**

```
printParameters()
```

---

putToEnv	<i>A function for changing and adding variables in the package environment.</i>
----------	---

---

**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	<i>A function for reading Bruker NMR spectra.</i>
------------	---

---

**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	<i>A function recreating parameters from previous runs.</i>
---------------	---

---

**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, graphics = TRUE)
```

**Arguments**

filename	File path/name of the mrbin parameter file to be loaded
graphics	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**Value**

None

**Examples**

```
resetEnv()#clean up previous data from the package environment
# Insert full folder path and file name
recreatemrbin(system.file("extdata/mrbin.txt",package="mrbin"))
```

---

removeFromPlot	<i>A function for removing NMR spectra from the plot list.</i>
----------------	--

---

**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	<i>A function for removing bins below noise level.</i>
-------------	--

---

**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**

```
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
  binwidth1D=0.05,noiseRemoval="No",PQNScaling="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",package="mrbin")))
results<-removeNoise(results)
```

---

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,
  graphics = TRUE
)
```

**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.
graphics	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**Value**

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

**Examples**

```

resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
  binwidth1D=0.05,PQNScaling="No",PCA="No",logTrafo="No",
  noiseRemoval="No",NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",package="mrbin")))
results<-removeSpectrum(results,
  spectra=c(system.file("extdata/2.mr1",package="mrbin")))

```

---

resetEnv	<i>A parameter resetting function</i>
----------	---------------------------------------

---

**Description**

This function resets the parameter variables in the package environment.

**Usage**

```
resetEnv()
```

**Value**

None

**Examples**

```
resetEnv()
```

---

right	<i>A function for changing plotNMR plots.</i>
-------	---

---

**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()#clean up previous data from the package environment
mrbin(silent=TRUE,parameters=list(dimension="1D",binwidth1D=.5,
  noiseRemoval="No",trimZeros="No",
  PQNScaling="No",saveFiles="No",referenceScaling="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",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, graphics = TRUE)
```

**Arguments**

spectrumNumber    If provided, this number will be used; defaults to NULL  
graphics           Controls whether pop-up windows are shown for selections. Defaults to TRUE.

**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,
  graphics = TRUE
)
```

### 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.

**graphics**     Controls whether pop-up windows are shown for selections. Defaults to TRUE.

### Value

An invisible mrbin object

### Examples

```
resetEnv()#clean up previous data from the package environment
results<-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,NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/2.mr1",package="mrbin"),
  system.file("extdata/3.mr1",package="mrbin"))
))
results<-setDilutionFactors(results,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,
  graphics = TRUE
)
```

### Arguments

<code>mrbinObject</code>	An mrbin object
<code>plotOnly</code>	Should only noise plots be generated (TRUE), or should noise levels be adjusted interactively (FALSE)
<code>showSpectrumPreview</code>	Should plots be shown? If not provided, this value will be taken from the mrbin object parameters
<code>silent</code>	If set to TRUE, plots will not be shown but might still be saved
<code>graphics</code>	Controls whether pop-up windows are shown for selections. Defaults to TRUE.

### Value

An invisible mrbin object

### Examples

```
resetEnv()#clean up previous data from the package environment
results<-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,NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/2.mr1",package="mrbin"),
  system.file("extdata/3.mr1",package="mrbin"))
))
results<-setNoiseLevels(results,plotOnly=TRUE)
```

---

setOffset	<i>A function for changing plotNMR plots.</i>
-----------	---

---

**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**

```
resetEnv()#clean up previous data from the package environment  
setOffset(0)
```

---

setParam	<i>A function setting parameters and checking for consistency.</i>
----------	--

---

**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**

```
resetEnv()#clean up previous data from the package environment  
setParam(parameters=list(dimension="1D"))
```

---

timeStampMrbin	<i>A function for time stamping mrbin objects.</i>
----------------	--

---

### 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

```
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D", logTrafo="No",
  binwidth1D=0.05,signal_to_noise1D=50, verbose=TRUE, PCA="No",
  trimZeros="No",NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"))))
results<-trimZeros(results)
```

---

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**

```
resetEnv()#clean up previous data from the package environment
results<-mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",
  binwidth1D=0.05,PQNScaling="No",PCA="No",logTrafo="No",
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=c(system.file("extdata/1.mr1",package="mrbin"),
    system.file("extdata/2.mr1",package="mrbin"),
    system.file("extdata/3.mr1",package="mrbin")))
results<-unitVarianceScaling(results)
```

---

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()#clean up previous data from the package environment
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",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,saveFiles="No",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr2",package="mrbin"),
  NMRvendor="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,
  dimension = "2D"
)

```

## Arguments

left	New left boundary
right	New right boundary
top	New top boundary
bottom	New bottom boundary
refreshPlot	Refresh plot automatically. Defaults to TRUE
dimension	Dimension of the data. Defaults to "2D"

## Value

An invisible value indicating if a change occurred

## Examples

```
resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",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

```
resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",package="mrbin")))
plotNMR()
zoomIn()
```

---

zoomOut	<i>A function for changing plotNMR plots.</i>
---------	---

---

**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**

```
resetEnv()#clean up previous data from the package environment
mrbin(silent=TRUE,setDefault=TRUE,parameters=list(dimension="1D",binwidth1D=.1,
  PQNScaling="No",noiseRemoval="No",trimZeros="No",
  fixNegatives="No",logTrafo="No",PCA="No",verbose=TRUE,
  NMRvendor="mrbin",
  example=TRUE,#only used for the package examples
  NMRfolders=system.file("extdata/1.mr1",package="mrbin")))
plotNMR()
zoomIn()
zoomOut()
```

# Index

addToPlot, 3  
annotatemrbin, 4  
atnv, 6

checkmrbin, 7  
contMin, 8  
contPlus, 8  
createAnnotationList, 9  
createmrbin, 10  
cropNMR, 10

dilutionCorrection, 11  
down, 12

editmetabolitesmrbin, 12  
editmrbin, 13

fia, 15

getEnv, 17

intMin, 17  
intPlus, 18

left, 19  
logTrafo, 19

metadatamrbin, 20  
mrbin, 21  
mrbinrun, 22  
mrheatmap, 23  
mrplot, 25

plotMultiNMR, 27  
plotNMR, 29  
plotPCA, 31  
plotResults, 32  
PQNScaling, 33  
predictWrapper, 35  
printParameters, 36  
putToEnv, 37

readBruker, 37  
recreatemrbin, 38  
removeFromPlot, 39  
removeNoise, 39  
removeSpectrum, 40  
resetEnv, 41  
right, 41

setCurrentSpectrum, 42  
setDilutionFactors, 43  
setNoiseLevels, 44  
setOffset, 45  
setParam, 45

timeStampMrbin, 46  
trimZeros, 47

unitVarianceScaling, 47  
up, 48

zoom, 49  
zoomIn, 50  
zoomOut, 51