

Package ‘MALDIquant’

May 6, 2012

Version 1.1

Date 2012-05-06

Title Quantitative Analysis of Mass Spectrometric Data

Author Sebastian Gibb

Maintainer Sebastian Gibb <mail@sebastiangibb.de>

Depends R (>= 2.10.0), methods

Description MALDIquant provides a complete analysis pipeline for MALDI-TOF and other mass spectrometric data. Distinctive features include baseline subtraction using the SNIP algorithm, peak alignment using warping functions, handling of replicated measurements as well as allowing spectra with different resolutions.

License GPL (>= 3)

URL <http://strimmerlab.org/software/malDIquant/>

LazyLoad yes

Repository CRAN

Date/Publication 2012-05-06 15:24:19

R topics documented:

MALDIquant-package	2
AbstractMassObject-class	4
binPeaks	5
calibrate-methods	7
createMassPeaks	8
createMassSpectrum	9
detectPeaks-methods	10
determineWarpingFunctions	12
estimateBaseline-methods	15
estimateNoise-methods	16

fiedler2009subset	17
filterPeaks	18
findEmptyMassObjects	20
intensityMatrix	21
isMassObject	22
isMassObjectList	23
labelPeaks-methods	24
MassPeaks-class	26
MassSpectrum-class	26
mergeMassSpectra	28
plot-methods	29
referencePeaks	31
removeBaseline-methods	32
standardizeTotalIonCurrent-methods	34
transformIntensity-methods	35
warpMassSpectra	36

Index	38
--------------	-----------

MALDIquant-package	<i>Quantitative Analysis of Mass Spectrometric Data</i>
--------------------	---

Description

MALDIquant provides a complete analysis pipeline for MALDI-TOF and other mass spectrometric data. Distinctive features include baseline subtraction using the SNIP algorithm, peak alignment using warping functions, handling of replicated measurements as well as allowing spectra with different resolutions.

For a first overview run `demo("MALDIquant")`.

Details

Package: MALDIquant
 Version: 1.1
 Date: 2012-05-06
 License: GPL (>= 3)
 URL: <http://strimmerlab.org/software/malDIquant/>

Main classes:

MassPeaks: Represents a peak list of a single spectrum.

MassSpectrum: Represents a single spectrum.

Author(s)

Sebastian Gibb

Maintainer: Sebastian Gibb <mail@sebastiangibb.de>

References

S. Gibb and K. Strimmer. 2012. *MALDIquant: a versatile R package for the analysis of mass spectrometric data*. <http://arxiv.org/abs/1203.5885>.

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[MassPeaks](#), [MassSpectrum](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## choose only a smaller subset
spectra <- fiedler2009subset[1:2];

## show some information
spectra

## running typical workflow

## transform intensities
t <- transformIntensity(spectra, fun=sqrt);

## smoothing function
movingAvg <- function(y) {
  ma <- rep(1, 5)/5;
  return(filter(y, ma, sides=2));
}

## smooth spectra
s <- transformIntensity(t, fun=movingAvg);

## baseline correction
b <- removeBaseline(s);

## detect peaks
p <- detectPeaks(b);

## plot results
par(mfcol=c(3, length(spectra)));
for (i in seq(along=spectra)) {
  plot(spectra[[i]], main="raw spectrum");
  plot(s[[i]], main="transformed (sqrt) and smoothed (ma5)");
  plot(b[[i]], main="baseline removed and peaks detected");
}
```

```

    points(p[[i]], pch=4, col=2);
  }
  par(mfrow=c(1, 1));

  ## bin peaks
  bp <- binPeaks(p);

  ## show intensity matrix
  intensityMatrix(bp);

```

AbstractMassObject-class

Class "AbstractMassObject"

Description

[AbstractMassObject](#) is an abstract (means pure virtual) class. It is the parent class of [MassSpectrum](#) and [MassPeaks](#). It shouldn't create or handle by the user because it is for internal use only.

Derived classes

[MassPeaks](#), [MassSpectrum](#)

Slots

.cache: environment, internal use only
mass: vector, mass or mass-to-charge ratio
intensity: vector, intensities for measured mass-to-charge ratios
metaData: list, some metadata to describe the spectrum

Methods

as.matrix signature(x = "AbstractMassObject"): Converts an [AbstractMassObject](#) object to a matrix with 2 columns (mass, intensity).

intensity signature(object = "AbstractMassObject"): Accessor function for slot intensity.

intensity<- signature(object = "AbstractMassObject", value = "numeric")
Replacement function for slot intensity.

isEmpty signature(object = "AbstractMassObject"): Returns TRUE if length of intensity is 0.

length signature(x = "AbstractMassObject"): Returns length of slot intensity.

lines signature(x = "AbstractMassObject"): Extended function for adding [AbstractMassObject](#) object as a line to a specific plot. See [lines](#) for details.

mass signature(object = "AbstractMassObject"): Accessor function for slot mass.

mass<- signature(object = "AbstractMassObject", value = "numeric")
Replacement function for slot mass.

metaData signature(object = "AbstractMassObject"): Accessor function for slot metaData.

metaData<- signature(object = "AbstractMassObject"): Replacement function for slot metaData.

plot signature(x = "AbstractMassObject", y = "missing"): Extended function for plotting an AbstractMassObject object. See [plot, AbstractMassObject, missing-method](#) for details.

points signature(x = "AbstractMassObject"): Extended function for adding [AbstractMassObject](#) object as points to a specific plot. See [points](#) for details.

totalIonCurrent signature(object = "AbstractMassObject"): Accessor function for sum of all intensities (slot intensity).

totalIonCurrent<- signature(object = "AbstractMassObject", value = "numeric")
Set sum of slot intensity to value.

transformIntensity signature(object = "AbstractMassObject"): Transforms the intensities of an AbstractMassObject object. See [transformIntensity, AbstractMassObject-method](#) for details.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldivant/>

See Also

[MassPeaks](#), [MassSpectrum](#), [plot, AbstractMassObject, missing-method](#), [transformIntensity, AbstractMassObject](#)

binPeaks

Align Peaks into discrete bins.

Description

This function looks for similar peaks (mass) across [MassPeaks](#) objects and equalizes their mass.

Usage

```
binPeaks(l, tolerance=0.002)
```

Arguments

l	list, list of MassPeaks objects.
tolerance	double, maximal deviation of a peak position (mass) to be considered as identical.

Details

The algorithm is based on the following workflow:

1. Put all mass in a sorted vector.
2. Calculate differences between each neighbor.
3. Divide the mass vector at the largest gap (largest difference) and form a left and a right bin.
4. Rerun step 3 for the left and/or the right bin if they don't fulfill the two criteria:
 - The bin doesn't contain two or more peaks of the same sample.
 - All peaks in a bin are near to the mean ($\text{abs}(\text{mass} - \text{meanMass}) / \text{meanMass} < \text{tolerance}$).

Value

Returns a [list](#) of mass adjusted [MassPeaks](#) objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[intensityMatrix](#), [MassPeaks](#)

Examples

```
## load library
library("MALDIquant");

## create two MassPeaks objects
p <- list(createMassPeaks(mass=seq(100, 500, 100),
                        intensity=1:5),
         createMassPeaks(mass=c(seq(100.2, 300.2, 100), 395),
                        intensity=1:4));

## only keep peaks which occur in all MassPeaks objects
binnedPeaks <- binPeaks(p, tolerance=0.002);

## compare result
iM1 <- intensityMatrix(p);
iM2 <- intensityMatrix(binnedPeaks);

all(dim(iM1) == c(2, 9)); # TRUE
all(dim(iM2) == c(2, 6)); # TRUE

show(iM2);

## increase tolerance
```

```
    binnedPeaks <- binPeaks(p, tolerance=0.1);  
    iM3 <- intensityMatrix(binnedPeaks);  
    all(dim(iM3) == c(2, 5)); # TRUE  
    show(iM3);
```

calibrate-methods *Calibrates intensity matrix.*

Description

This function calibrates intensities of an intensity matrix created by [intensityMatrix](#).

Usage

```
calibrate(x)
```

Arguments

`x` `matrix`, an intensity matrix created by [intensityMatrix](#). (Rows have to contain samples and columns features (peaks).)

Details

First a *median spectrum/feature list* is calculated to be used as reference. Afterwards each sample is divided by the median of the ratios of sample intensities to reference intensities.

Value

Returns a calibrated `matrix` (scale/calibration factors as attribute).

Author(s)

Sebastian Gibb

See Also

[AbstractMassObject](#),

Examples

```
## load library
library("MALDIquant");

## create example matrix
m <- matrix(c(1, 2, 3,
              3, 6, 9),
            nrow=2, byrow=TRUE, dimnames=list(paste("samples", 1:2, sep=""),
                                              paste("peaks", LETTERS[1:3])));

## show matrix
m

## show calibrated matrix
calibrate(m);
```

createMassPeaks	<i>Creates a MassPeaks object.</i>
-----------------	------------------------------------

Description

This function creates a [MassPeaks](#) object. Normally it shouldn't be called by the user. Try [detectPeaks](#), [MassSpectrum-method](#) instead.

Usage

```
createMassPeaks(mass, intensity, metaData=list())
```

Arguments

mass	vector, mass or mass-to-charge ratio
intensity	vector, intensities for measured mass-to-charge ratios
metaData	list, some metadata to describe the spectrum

Value

Returns a [MassPeaks](#) object.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[detectPeaks](#), [MassSpectrum-method](#)

Examples

```
## load library
library("MALDIquant");

## create a MassPeaks object by default constructor
s <- createMassPeaks(mass=1:100, intensity=rnorm(100)^2,
                    metaData=list(name="example peaks"));

## show some details
s;
```

createMassSpectrum *Creates a MassSpectrum object.*

Description

This function creates a [MassSpectrum](#) object.

Usage

```
createMassSpectrum(mass, intensity, metaData=list())
```

Arguments

mass	vector, mass or mass-to-charge ratio
intensity	vector, intensities for measured mass-to-charge ratios
metaData	list, some metadata to describe the spectrum

Value

Returns a [MassSpectrum](#) object.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

Examples

```
## load library
library("MALDIquant");

## create a MassSpectrum object by default constructor
s <- createMassSpectrum(mass=1:100, intensity=rnorm(100)^2,
                       metaData=list(name="example spectrum"));

## show some details
s;
```

detectPeaks-methods *Detects peaks in a MassSpectrum object.*

Description

This method looks for peaks in mass spectrometry data (represented by a [MassSpectrum](#) object). A peak is a local maximum above a user defined noise threshold.

Usage

```
## S4 method for signature 'MassSpectrum'
detectPeaks(object,
            halfWindowSize=20, fun, SNR=2,
            ...)
```

Arguments

object	MassSpectrum object or a list of MassSpectrum objects.
halfWindowSize	numeric, half window size. The resulting window reaches from <code>mass[currentIndex-halfWindowSize]</code> to <code>mass[currentIndex+halfWindowSize]</code> . A local maximum have to be the highest one in the given window to be recognized as peak.
fun	function, noise estimation function. If no fun argument is given the noise would be estimated automatically by calling estimateNoise,MassSpectrum-method .
SNR	single numeric value. SNR is an abbreviation for signal-to-noise-ratio. A local maximum has to be higher than $SNR \times noise$ to be recognize as peak.
...	arguments to be passed to fun

Details

fun: A user-defined function to estimate the noise of a [MassSpectrum](#) object. At least three arguments are needed:

- x: double, the mass of the [MassSpectrum](#) object.
- y: double, the intensity of the [MassSpectrum](#) object.

- ...: further arguments

A correct noise estimation function has to return a two column matrix with mass (x-values) as first and intensities (y-values) as second column or a single numeric value.

```
## e.g. estimate noise using Friedman's 'super smoother'  
estimateNoiseSuperSmoother <- function(x, y, ...) {  
  s <- supsmu(x=x, y=y, ...);  
  
  return(cbind(x, s$y));  
}
```

Value

Returns a [MassPeaks](#) object.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[MassPeaks](#), [MassSpectrum](#), [estimateNoise](#), [MassSpectrum-method](#)

Examples

```
## load library  
library("MALDIquant");  
  
## load example data  
data("fiedler2009subset", package="MALDIquant");  
  
## choose only the first mass spectrum  
s <- fiedler2009subset[[1]];  
  
## transform intensities  
s <- transformIntensity(s, sqrt);  
  
## remove baseline  
s <- removeBaseline(s);  
  
## plot spectrum  
plot(s);  
  
## call peak detection  
p <- detectPeaks(s);
```

```
## draw peaks on the plot
points(p);

## label 10 highest peaks
top10 <- intensity(p) %in% sort(intensity(p), decreasing=TRUE)[1:10];
labelPeaks(p, index=top10);
```

determineWarpingFunctions

Determine warping functions of MassPeaks objects.

Description

This function determines a warping function for a list of [AbstractMassObject](#) objects (warping is also known as *phase correction/spectra alignment*).

Usage

```
determineWarpingFunctions(l, reference, tolerance=0.002,
                          warpingFunction=.warpingFunctionLowess,
                          plot=FALSE, plotInteractive=FALSE, ...)
```

Arguments

<code>l</code>	list, list of MassPeaks objects.
<code>reference</code>	MassPeaks , reference object to which the samples (<code>l</code>) should be aligned. If missing referencePeaks is used.
<code>tolerance</code>	double, maximal deviation of a peak position (mass) to be considered as identical.
<code>warpingFunction</code>	function, general warping function which is used to align sample and reference MassPeaks objects.
<code>plot</code>	logical, if TRUE a warping plot is drawn for each sample.
<code>plotInteractive</code>	logical, if FALSE a non-interactive device (e.g. pdf) is used for warping plots.
<code>...</code>	arguments to be passed to <code>warpingFunction</code>

Details

`warpingFunction`: `determineWarpingFunctions` estimates a warping function to overcome the difference between mass in reference and in the current sample. To calculate the differences each reference peak would match with the highest sample peak in the nearer neighborhood (defined by mass of reference peak*`tolerance`).

MALDIquant uses a `lowess`-based warping function as default one. It could easily be replaced by an own warping function. At least three arguments are needed:

- x: double, the original (not warped) mass of the sample `MassPeaks` object.
- d: double, the difference of the sample and the reference mass.
- ...: further arguments (e.g. `iter` for `lowess`)

A correct warping function has to return a single individual warping function which depends on a single argument x.

```
## e.g. warping function to fit a 2nd order polynomial
quadraticWarp <- function(x, d, ...) {
  l <- lm(y ~ x1+x2, data=list(x1=x, x2=x*x, y=d), ...);
  co <- coef(l);
  return(function(x) { return (co[1]+x*co[2]+x*x*co[3]) });
}
```

`plotInteractive`: If `plot` is `TRUE` a lot of output is created (each sample in `l` gets its own plot). That's why a non-interactive device is recommended:

```
## create a device
pdf()
## calculate warping functions
w <- determineWarpingFunctions(p, plot=TRUE)
## close device
dev.off();
```

Value

Returns a list of individual warping functions.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[referencePeaks](#), [warpMassPeaks](#), [warpMassSpectra](#), [MassPeaks](#)

Examples

```
## load library
library("MALDIquant");

## create a reference MassPeaks object
r <- createMassPeaks(mass=1:5, intensity=1:5);
```

```
## create test samples
p <- list(createMassPeaks(mass=((1:5)*1.01), intensity=1:5),
          createMassPeaks(mass=((1:5)*0.99), intensity=1:5));

## create a simple (in other cases useless ) warping function
simpleWarp <- function(x, d, ...) { return(function(x){return(d)}); }

## create an interactive device with 2 rows
par(mfrow=c(2, 1));
## calculate warping function (using simpleWarp as basic warping function)
## and show warping plot
w <- determineWarpingFunctions(p, tolerance=0.02, warpingFunction=simpleWarp,
                              plot=TRUE, plotInteractive=TRUE);
par(mfrow=c(1, 1));

## w contains the individual warping functions
warpedPeaks <- warpMassPeaks(p, w);

## compare results
all(mass(r) == mass(warpedPeaks[[1]])); # TRUE
all(mass(r) == mass(warpedPeaks[[2]])); # TRUE

## realistic example

## load example data
data("fiedler2009subset", package="MALDIquant");

## running typical workflow

## transform intensities
t<- transformIntensity(fiedler2009subset, fun=sqrt);

## smoothing function
movingAvg <- function(y) {
  return(filter(y, rep(1, 5)/5, sides=2));
}

## smooth spectra
s <- transformIntensity(t, fun=movingAvg);

## baseline correction
b <- removeBaseline(s);

## detect peaks
peaks <- detectPeaks(b);

## create an interactive device with 2 rows
par(mfrow=c(2, 1));
## calculate warping functions (using lowess based basic function [default])
w <- determineWarpingFunctions(peaks, plot=TRUE, plotInteractive=TRUE);
```

```
par(mfrow=c(1, 1));
```

estimateBaseline-methods

Estimates the baseline of a MassSpectrum object.

Description

This method estimates the baseline of mass spectrometry data (represented by a [MassSpectrum](#) object).

Usage

```
## S4 method for signature 'MassSpectrum'  
estimateBaseline(object,  
  method=c("SNIP", "ConvexHull", "Median"),  
  ...)
```

Arguments

object	MassSpectrum object
method	a baseline estimation function; see .estimateBaselineSnip , .estimateBaselineConvexHull , .estimateBaselineMedian
...	arguments to be passed to method

Value

Returns a two column matrix (first column: mass, second column: intensity) of the estimated baseline.

Author(s)

Sebastian Gibb

See Also

[MassSpectrum](#), [.estimateBaselineSnip](#), [.estimateBaselineConvexHull](#), [.estimateBaselineMedian](#), [removeBaseline](#), [MassSpectrum-method](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## choose only the first mass spectrum
s <- fiedler2009subset[[1]];

## plot spectrum
plot(s);

## estimate baseline
b <- estimateBaseline(s);

## draw baseline on the plot
lines(b, col="red");
```

estimateNoise-methods *Estimates the noise of a MassSpectrum object.*

Description

This method estimates the noise of mass spectrometry data (represented by a [MassSpectrum](#) object) by calculating the median absolute deviation.

Usage

```
## S4 method for signature 'MassSpectrum'
estimateNoise(object)
```

Arguments

object [MassSpectrum](#) object

Value

Returns a single numeric value.

Author(s)

Sebastian Gibb

See Also

[MassSpectrum](#), [detectPeaks](#), [MassSpectrum-method](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## choose only the first mass spectrum
s <- fiedler2009subset[[1]];

## transform intensities
s <- transformIntensity(s, sqrt);

## remove baseline
s <- removeBaseline(s);

## plot spectrum
plot(s);

## estimate noise
n <- estimateNoise(s);

## draw noise on the plot
abline(h=n, col="red");
```

fiedler2009subset *Example Mass Spectra (raw)*

Description

This dataset contains 16 example mass spectra. It is used to demonstrate the usage of [MALDIquant-package](#).

Usage

```
fiedler2009subset
```

Format

A list containing 16 [MassSpectrum-class](#) objects.

Details

The dataset is a subset of data used in *Fiedler et al 2009*.

It contains spectra of 8 different patients (each one has 2 technical replicates).

list_index	laboratory	patient_id	sex	age	type
1	Leipzig	LC77	male	37	control
2	Leipzig	LC77	male	37	control
3	Leipzig	LC213	female	51	control

4	Leipzig	LC213	female	51	control
5	Leipzig	LT178	male	58	cancer
6	Leipzig	LT178	male	58	cancer
7	Leipzig	LT157	male	60	cancer
8	Leipzig	LT157	male	60	cancer
9	Heidelberg	HC49	male	43	control
10	Heidelberg	HC49	male	43	control
11	Heidelberg	HC54	female	71	control
12	Heidelberg	HC54	female	71	control
13	Heidelberg	HT151	male	53	cancer
14	Heidelberg	HT151	male	53	cancer
15	Heidelberg	HT429	female	58	cancer
16	Heidelberg	HT429	female	58	cancer

References

G.M. Fiedler, A.B. Leichtle, J. Kase, S. Baumann, U. Ceglarek, K. Felix, T. Conrad, H. Witzigmann, A. Weimann, C. Schütte, J. Hauss, M. Büchler and J. Thiery
 “Serum Peptidome Profiling Revealed Platelet Factor 4 as a Potential Discriminating Peptide Associated with Pancreatic Cancer”
 Clinical Cancer Research, 11(15): 3812-3819, 2009
 ISSN 1557-3265; doi:10.1158/1078-0432.CCR-08-2701
<http://clincancerres.aacrjournals.org/content/15/11/3812>

See Also

[MassSpectrum-class](#)

filterPeaks	<i>Removes less frequently peaks.</i>
-------------	---------------------------------------

Description

This function removes infrequently peaks from a list of [MassPeaks](#) objects.

Usage

```
filterPeaks(l, minFrequency, labels)
```

Arguments

l	list, list of MassPeaks objects.
minFrequency	double, remove all peaks which occur in less than minFrequency*length(l) MassPeaks objects.
labels	list, list of factor s (one for each MassPeaks object) to do groupwise filtering.

Value

Returns a [list](#) of filtered [MassPeaks](#) objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[intensityMatrix](#), [MassPeaks](#)

Examples

```
## load library
library("MALDIquant");

## create four MassPeaks objects and add them to the list
p <- list(createMassPeaks(mass=1:2, intensity=1:2),
          createMassPeaks(mass=1:3, intensity=1:3),
          createMassPeaks(mass=1:4, intensity=1:4),
          createMassPeaks(mass=1:5, intensity=1:5))

## only keep peaks which occur in all MassPeaks objects
filteredPeaks <- filterPeaks(p, minFrequency=1);

## compare result
intensities <- intensityMatrix(filteredPeaks);

## peaks at mass 3,4,5 are removed
all(dim(intensities) == c(4, 2)); # TRUE
all(intensities[,1] == 1);      # TRUE
all(intensities[,2] == 2);      # TRUE

## only keep peaks which occur in all MassPeaks in a group
## (e.g. useful for technical replicates)
groups <- as.factor(c("a", "a", "b", "b"));
filteredPeaks <- filterPeaks(p, minFrequency=1, labels=groups);

## peaks at mass 3 removed in group "a"
filteredPeaks[groups == "a"];

## peaks at mass 5 removed in group "b"
filteredPeaks[groups == "b"];
```



```
peakList[[2]] <- createMassPeaks(mass=1:100, intensity=1:100,
                                metaData=list(name="example 2"));

## find empty objects (there should not be any one)
findEmptyMassObjects(peakList);

## add an empty MassPeaks object to the list
peakList[[3]] <- createMassPeaks(mass=double(), intensity=double(),
                                metaData=list(name="empty MassPeaks object"));

## look for empty objects (isEmptyIdx == 3)
(isEmptyIdx <- findEmptyMassObjects(peakList));

## to remove all empty MassObjects from a list
length(peakList); ## 3
peakList <- removeEmptyMassObjects(peakList);
length(peakList); ## 2; WARNING: all indices could changed
```

intensityMatrix	<i>Converts a list MassObject objects into a matrix.</i>
-----------------	--

Description

This function converts a [list](#) of [AbstractMassObject](#) objects into a [matrix](#).

Usage

```
intensityMatrix(l)
```

Arguments

1 list, list of [AbstractMassObject](#) objects.

Details

The column names of the returned [matrix](#) store the mass values.

Value

Returns a [matrix](#) containing intensities of all [AbstractMassObject](#) objects of `l`. The [matrix](#) has `length(l)` rows (one row for each sample) and `length(unique(mass))` columns.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[AbstractMassObject](#),

Examples

```
## load library
library("MALDIquant");

## create example MassPeaks objects
p <- list(createMassPeaks(mass=1:4,
                        intensity=11:14,
                        metaData=list(name="test mass peaks 1")),
         createMassPeaks(mass=2:5,
                        intensity=22:25,
                        metaData=list(name="test mass peaks 2")));

## converts MassPeaks objects into a matrix
intensityMatrix(p);
```

isMassObject	<i>Tests for AbstractMassObject objects.</i>
--------------	--

Description

These functions test for an [AbstractMassObject](#) object.

Usage

```
isMassObject(x)

isMassSpectrum(x)

isMassPeaks(x)
```

Arguments

x object to be tested.

Value

Returns **TRUE** or **FALSE** depending on whether its argument is an [AbstractMassObject](#) object.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[MassPeaks](#), [MassSpectrum](#), [AbstractMassObject](#)

Examples

```
## load library
library("MALDIquant");

## create a MassPeaks object
peaks <- createMassPeaks(mass=1:100, intensity=1:100,
                        metaData=list(name="example 1"));

## test
isMassObject(peaks); # returns TRUE
isMassPeaks(peaks); # returns TRUE
isMassSpectrum(peaks); # returns FALSE
isMassPeaks(double()); # returns FALSE
```

isMassObjectList	<i>Tests a list of AbstractMassObject objects.</i>
------------------	--

Description

These functions test a [list](#) whether containing [AbstractMassObject](#) objects.

Usage

```
isMassObjectList(x)

isMassSpectrumList(x)

isMassPeaksList(x)
```

Arguments

x object to be tested.

Value

Returns [TRUE](#) or [FALSE](#) depending on whether its argument is a [list](#) of [AbstractMassObject](#) objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[MassPeaks](#), [MassSpectrum](#), [AbstractMassObject](#)

Examples

```
## load library
library("MALDIquant");

## create list
p <- list();

## test list
isMassPeaksList(p); # returns FALSE

## create two MassPeaks objects and add them to the list
p <- createMassPeaks(mass=1:100, intensity=1:100,
  metaData=list(name="example 1"));
p <- createMassPeaks(mass=1:100, intensity=1:100,
  metaData=list(name="example 2"));

## test list
isMassObjectList(p); # returns TRUE
isMassPeaksList(p); # returns TRUE
isMassSpectrumList(p); # returns FALSE
```

labelPeaks-methods *Draws peak labels to plot.*

Description

[labelPeaks](#) draws the corresponding mass values on top of the peaks stored in a [MassPeaks](#) object to a plot.

Usage

```
## S4 method for signature 'MassPeaks'
labelPeaks(object,
  index,
  mass, tolerance=0.002,
  digits=3, underline=TRUE,
  verticalOffset=abs(diff(par("usr")[3:4]))*0.0125,
  absoluteVerticalPos,
  adj=c(0.5, 0), cex=0.7, family="sans",
  ...)
```

Arguments

object	MassPeaks object.
index	vector, indices of peaks to label.
mass	vector, mass of peaks to label.
tolerance	double, maximal deviation of a peak position (mass) to be considered.
digits	integer, number of decimal places.
underline	logical, underline peak values?
verticalOffset	numeric, move label vertically (relative to peak height).
absoluteVerticalPos	numeric, absolute y value for the label. If missing verticalOffset is used.
adj	numeric, adjust text to the left, center, right and top, center, bottom; see text .
cex	numeric, font size, see par .
family	name of font family, see par .
...	arguments to be passed to text .

Author(s)

Sebastian Gibb

See Also

[MassPeaks](#), [plot](#), [AbstractMassObject](#), [missing-method](#)

Examples

```
## load library
library("MALDIquant");

## create a MassPeaks object from scratch
p <- createMassPeaks(mass=1:20, intensity=sample(x=100:10000, size=20),
                    metaData=list(name="example"));

## plot peaks
plot(p);

## label the first 5 peaks
labelPeaks(p, index=1:5);

## label all peaks in mass range 15 to 20
labelPeaks(p, mass=15:20, underline=FALSE);

## label highest peaks (top 5)
top5 <- intensity(p) %in% sort(intensity(p), decreasing=TRUE)[1:5];
labelPeaks(p, index=top5, col="red");
```

MassPeaks-class *Class "MassPeaks"*

Description

[MassPeaks](#) represents extracted peaks of a single spectrum of a MALDI/TOF mass spectrometry measurement.

Objects from the Class

[createMassPeaks](#): Creates a [MassPeaks](#) object.

Extends

Class [AbstractMassObject](#), directly.

Methods

labelPeaks signature(x = "MassPeaks"): Draws peak labels to plot. See [labelPeaks, MassPeaks-method](#) for details.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[createMassPeaks](#), [labelPeaks](#), [MassPeaks-method](#), [AbstractMassObject](#)

MassSpectrum-class *Class "MassSpectrum"*

Description

[MassSpectrum](#) represents a single spectrum of a MALDI-TOF mass spectrometry measurement. It provides an easy framework for doing some preprocessing steps like peak detection, baseline correction and much more.

Objects from the Class

[createMassSpectrum](#): Creates a [MassSpectrum](#) object.

Extends

Class [AbstractMassObject](#), directly.

Methods

detectPeaks signature(x = "MassSpectrum"): Look for local maxima and estimate noise to extract peaks out of a [MassSpectrum](#) object. See [detectPeaks,MassSpectrum-method](#) for details.

estimateBaseline signature(x = "MassSpectrum"): Estimates the baseline of a [MassSpectrum](#) object. See [estimateBaseline,MassSpectrum-method](#) for details.

estimateNoise signature(x = "MassSpectrum"): Estimates the noise of a [MassSpectrum](#) object. See [estimateNoise,MassSpectrum-method](#) for details.

removeBaseline signature(x = "MassSpectrum"): Estimates and removes the baseline of a [MassSpectrum](#) object. See [removeBaseline,MassSpectrum-method](#) for details.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[createMassSpectrum](#), [detectPeaks,MassSpectrum-method](#), [estimateBaseline,MassSpectrum-method](#), [estimateNoise,MassSpectrum-method](#), [imputeMass,MassSpectrum-method](#), [removeBaseline,MassSpectrum-method](#), [AbstractMassObject](#)

Examples

```
## load library
library("MALDIquant");

## create a MassSpectrum object by default constructor
s <- createMassSpectrum(mass=1:100, intensity=rnorm(100)^2,
                       metaData=list(name="example"));

## show some details
s;

## plot spectrum
plot(s);

## modify intensity and metaData
intensity(s)[1:50] <- 0;
metaData(s) <- list(name="modified example");

## plot again
plot(s);
```

mergeMassSpectra *Merges MassPeaks or MassSpectrum objects.*

Description

These functions merge [MassPeaks](#) or [MassSpectrum](#) objects by a user defined function.

Usage

```
mergeMassPeaks(l, labels, fun=mean, ...)
```

```
mergeMassSpectra(l, labels, fun=mean, ...)
```

Arguments

l	list, list of MassPeaks or MassSpectrum objects.
labels	list, list of factor s (one for each MassPeaks object) to do groupwise merging.
fun	used merge function (mean , median , sum or something else would be possible).
...	arguments to be passed to fun

Value

Returns a single (no labels given) or a [list](#) (labels given) of merged [MassPeaks](#) or [MassSpectrum](#) objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[MassPeaks](#), [MassSpectrum](#)

Examples

```
## load library
library("MALDIquant");

## create four MassPeaks objects and add them to the list
p<- list(createMassPeaks(mass=1:2, intensity=1:2),
         createMassPeaks(mass=1:3, intensity=1:3),
         createMassPeaks(mass=1:4, intensity=1:4),
         createMassPeaks(mass=1:5, intensity=1:5))
```

```

## merge all four MassPeaks objects into a single new one
## by sum their intensities
## (no labels, returns only one new MassPeaks object)
mergedPeaks <- mergeMassPeaks(p, fun=sum);

## compare result
mass(mergedPeaks)      # 1:5
intensity(mergedPeaks) # c(4, 8, 9, 8, 5)

## only merge MassPeaks objects in a group
## (e.g. useful for technical replicates)
## (two different labels, returns a list of two new MassPeaks objects)
groups <- as.factor(c("a", "a", "b", "b"));
mergedPeaks <- mergeMassPeaks(p, labels=groups, fun=sum);

## compare result
## group "a":
mass(mergedPeaks[[1]]) # 1:3
intensity(mergedPeaks[[1]]) # c(2, 4, 3)

## group "b":
mass(mergedPeaks[[2]]) # 1:5
intensity(mergedPeaks[[2]]) # c(2, 4, 6, 8, 5)

```

plot-methods

Plots an AbstractMassObject object.

Description

This is an overloaded method to allow plotting of an [AbstractMassObject](#) object.

Usage

```

## S4 method for signature 'AbstractMassObject,missing'
plot(x, col="black",
     xlab="mass", ylab="intensity",
     type=ifelse(isMassPeaks(x), "h", "l"),
     xlim=c(ifelse(length(x@mass)>0, min(x@mass, na.rm=TRUE), 0),
            ifelse(length(x@mass)>0, max(x@mass, na.rm=TRUE), 1)),
     ylim=c(0, ifelse(length(x@intensity)>0, max(x@intensity, na.rm=TRUE), 1)),
     main=x@metaData$name, sub=x@metaData$file,
     cex.sub=0.75, col.sub="#808080",
     abline.col="#808080", ...)

```

Arguments

x	MassSpectrum object
col	line colour, see par

xlab	title for the x-axis, see title
ylab	title for the y-axis, see title
type	type of plot: see plot
xlim	the x limits (x1, x2) of the plot, see plot.default
ylim	the y limits (y1, y2) of the plot, see plot.default
main	title for the plot, see title
sub	sub title for the plot, see title
cex.sub	sub title font size, see par
col.sub	sub title color, see par
abline.col	colour for horizontal line at y=0
...	arguments to be passed to plot

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[AbstractMassObject](#),

Examples

```
## load library
library("MALDIquant");

## create a MassSpectrum object by default constructor
s <- createMassSpectrum(mass=1:100, intensity=rnorm(100)^2,
                       metaData=list(name="example"));

## show some details
s;

## plot spectrum
plot(s);
```

referencePeaks	Creates a reference <code>MassPeaks</code> object.
----------------	--

Description

This function creates a reference `MassPeaks` object (also called *Anchor Peaks*) from a list of `MassPeaks` objects.

Generally it is a combination of `binPeaks` and `filterPeaks`

Usage

```
referencePeaks(l, minFrequency=0.9, tolerance=0.002)
```

Arguments

<code>l</code>	list, list of <code>MassPeaks</code> objects.
<code>minFrequency</code>	double, remove all peaks which occur in less than <code>minFrequency*length(l)</code> <code>MassPeaks</code> objects.
<code>tolerance</code>	double, maximal deviation of a peak position (mass) to be considered as identical.

Value

Returns a new `MassPeaks` objects.

The `intensity` slot of the returned `MassPeaks` represents the frequency of this mass position in all samples.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[binPeaks](#), [filterPeaks](#), [MassPeaks](#)

Examples

```
## load library
library("MALDIquant");

## create four MassPeaks objects and add them to the list
p<- list(createMassPeaks(mass=1:2, intensity=1:2),
         createMassPeaks(mass=1:3, intensity=1:3),
         createMassPeaks(mass=1:4, intensity=1:4),
```

```

createMassPeaks(mass=1:5, intensity=1:5))

## only use peaks which occur in all MassPeaks objects as reference peaks
refPeaks <- referencePeaks(p, minFrequency=1);

mass(refPeaks)      # 1:2
intensity(refPeaks) # c(1, 1)

```

removeBaseline-methods

Removes the baseline of a MassSpectrum object.

Description

This method removes the baseline of mass spectrometry data (represented by a [MassSpectrum](#) object).

The intensity of the mass spectrometry data would be reduced by baseline.

Usage

```

## S4 method for signature 'MassSpectrum'
removeBaseline(object,
  fun,
  ...)

```

Arguments

object	MassSpectrum object or a list of MassSpectrum objects.
fun	function, baseline estimation function. If no fun argument is given the baseline would be estimate automatically by calling estimateBaseline,MassSpectrum-method .
...	arguments to be passed to fun or to estimateBaseline,MassSpectrum-method .

Details

fun: A user-defined function to estimate the baseline of a [MassSpectrum](#) object.

At least three arguments are needed:

- x: double, the mass of the [MassSpectrum](#) object.
- y: double, the intensity of the [MassSpectrum](#) object.
- ...: further arguments

A correct baseline estimation function has to return a two column matrix with mass (x-values) as first and intensities (y-values) as second column.

```
## e.g. moving median baseline estimation
estimateBaselineMovingMedian <- function(x, y, halfWindowSize=100) {
  if (halfWindowSize<1) {
    stop(sQuote("halfWindowSize"), "=", halfWindowSize, " is too small!");
  }

  m <- runmed(y, k=(2*halfWindowSize+1));

  return(cbind(x, m));
}
```

Value

Returns a modified [MassSpectrum](#) object with reduced intensities.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[MassSpectrum](#), [estimateBaseline](#), [MassSpectrum-method](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## choose only the first mass spectrum
s <- fiedler2009subset[[1]];

## plot spectrum
plot(s);

## subtract baseline
b <- removeBaseline(s);

## draw modified spectrum on the plot
lines(b, col="blue");
```

standardizeTotalIonCurrent-methods

Standardizes the Total Ion Current of a list of AbstractMassObject objects.

Description

This method sets the TIC (*Total Ion Current*) of a list of `AbstractMassObject` objects to one. That results in a decreased intensity of the mass spectrometry data.

Usage

```
## S4 method for signature 'list'  
standardizeTotalIonCurrent(object,  
  value=1)
```

Arguments

object	a list of <code>AbstractMassObject</code> objects
value	numeric, resulting TIC

Details

`standardizeTotalIonCurrent` is a shortcut for:

```
stic <- lapply(spectralList, "totalIonCurrent<=", 1);
```

Value

Returns a list of modified `AbstractMassObject` objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[AbstractMassObject](#), [MassSpectrum](#), [MassPeaks](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## show TIC in spectrum 1
totalIonCurrent(fiedler2009subset[[1]]) ## 90312326

## set TIC to 1 in all spectra
sTIC <- standardizeTotalIonCurrent(fiedler2009subset);

## show TIC in spectrum 1
totalIonCurrent(sTIC[[1]]) ## 1
```

transformIntensity-methods

Transforms Intensities of an AbstractMassObject object.

Description

This method performs a transformation (e.g. sqrt-transformation) on the intensities of an [AbstractMassObject](#) object.

Usage

```
## S4 method for signature 'AbstractMassObject'
transformIntensity(object,
  fun, na.rm = TRUE,
  ...)
```

Arguments

object	AbstractMassObject object or a list of AbstractMassObject objects.
fun	used transformation function (sqrt , log10 , log or something else would be possible)
na.rm	logical, indicating whether 'NA' values should be removed
...	arguments to be passed to fun

Details

[transformIntensity](#) is a shortcut for transforming intensities manually.

```
## doing things by hand
newMassSpectrum <- singleSpectrum1;
intensity(newMassSpectrum) <- sqrt(intensity(singleSpectrum1));

## the code above could be replaced by the following line
newMassSpectrum <- transformIntensity(singleSpectrum1, fun=sqrt);
```

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/malDIquant/>

See Also

[AbstractMassObject](#), [MassSpectrum](#)

Examples

```
## load library
library("MALDIquant");

## load example data
data("fiedler2009subset", package="MALDIquant");

## choose only the first mass spectrum
s <- fiedler2009subset[[1]];

## transform spectrum
t <- transformIntensity(s, fun=sqrt);

## plot spectrum
par(mfrow=c(2, 1));
plot(s, main="raw spectrum");
plot(t, main="transformed spectrum");
par(mfrow=c(1, 1));
```

warpMassSpectra

Run warping functions on AbstractMassObject objects.

Description

These functions run warping functions on [AbstractMassObject](#) objects (warping is also known as *phase correction*).

Usage

```
warpMassPeaks(l, w)
```

```
warpMassSpectra(l, w)
```

Arguments

l list, list of [MassPeaks](#) or [MassSpectrum](#) objects.

w a list of warping functions determined by [determineWarpingFunctions](#). Has to be of the same length as **l**.

Details

The warping function *w* is called in the following way:

$$newMass = oldMass + w(oldMass)$$

Value

Returns a list of warped [MassPeaks](#) or [MassSpectrum](#) objects.

Author(s)

Sebastian Gibb

References

See website: <http://strimmerlab.org/software/maldiquant/>

See Also

[determineWarpingFunctions](#), [MassPeaks](#), [MassSpectrum](#)

Examples

```
## load library
library("MALDIquant");

## create a MassPeaks object
p <- createMassPeaks(mass=1:5, intensity=1:5);

## simple warping function (see determineWarpingFunctions for details)
simpleWarp <- function(x) { return(1); }

## run warping function
w <- warpMassPeaks(list(p), list(simpleWarp))[[1]];

## compare results
all(mass(w) == mass(p)+1); # TRUE
```

Index

*Topic **classes**

AbstractMassObject-class, 4
MassPeaks-class, 26
MassSpectrum-class, 26

*Topic **datasets**

fiedler2009subset, 17

*Topic **methods**

binPeaks, 5
calibrate-methods, 7
createMassPeaks, 8
createMassSpectrum, 9
detectPeaks-methods, 10
determineWarpingFunctions, 12
estimateBaseline-methods, 15
estimateNoise-methods, 16
filterPeaks, 18
findEmptyMassObjects, 20
intensityMatrix, 21
isMassObject, 22
isMassObjectList, 23
labelPeaks-methods, 24
mergeMassSpectra, 28
plot-methods, 29
referencePeaks, 31
removeBaseline-methods, 32
standardizeTotalIonCurrent-methods, 34
transformIntensity-methods, 35
warpMassSpectra, 36

*Topic **package**

MALDIquant-package, 2
.estimateBaselineConvexHull, 15
.estimateBaselineMedian, 15
.estimateBaselineSnip, 15

AbstractMassObject, 4, 5, 7, 12, 20–24, 26, 27, 29, 30, 34–36
AbstractMassObject-class, 4
as.matrix, AbstractMassObject-method (AbstractMassObject-class), 4

binPeaks, 5, 31

calibrate (calibrate-methods), 7
calibrate, matrix-method (calibrate-methods), 7

calibrate-methods, 7
createMassPeaks, 8, 26
createMassSpectrum, 9, 26, 27

detectPeaks (detectPeaks-methods), 10
detectPeaks, list-method (detectPeaks-methods), 10
detectPeaks, MassSpectrum-method (detectPeaks-methods), 10
detectPeaks-methods, 10
determineWarpingFunctions, 12, 37

estimateBaseline (estimateBaseline-methods), 15
estimateBaseline, MassSpectrum-method (estimateBaseline-methods), 15
estimateBaseline-methods, 15
estimateNoise (estimateNoise-methods), 16
estimateNoise, MassSpectrum-method (estimateNoise-methods), 16
estimateNoise-methods, 16

factor, 18, 28
FALSE, 22, 23
fiedler2009subset, 17
filterPeaks, 18, 31
findEmptyMassObjects, 20

intensity, 31
intensity (AbstractMassObject-class), 4
intensity, AbstractMassObject-method (AbstractMassObject-class), 4
intensity<- (AbstractMassObject-class), 4

- intensity<- ,AbstractMassObject,numeric-method
 - mergeMassSpectra, 28
 - (AbstractMassObject-class), 4
- intensityMatrix, 6, 7, 19, 21
- isEmpty (AbstractMassObject-class), 4
- isEmpty, AbstractMassObject-method
 - (AbstractMassObject-class), 4
- isMassObject, 22
- isMassObjectList, 23
- isMassPeaks (isMassObject), 22
- isMassPeaksList (isMassObjectList), 23
- isMassSpectrum (isMassObject), 22
- isMassSpectrumList (isMassObjectList), 23

- labelPeaks, 24
- labelPeaks (labelPeaks-methods), 24
- labelPeaks, MassPeaks-method
 - (labelPeaks-methods), 24
- labelPeaks-methods, 24
- length, AbstractMassObject-method
 - (AbstractMassObject-class), 4
- lines, 4
- lines, AbstractMassObject-method
 - (AbstractMassObject-class), 4
- list, 6, 19–21, 23, 28
- log, 35
- log10, 35
- lowess, 12, 13

- MALDIquant (MALDIquant-package), 2
- MALDIquant-package, 2
- mass (AbstractMassObject-class), 4
- mass, AbstractMassObject-method
 - (AbstractMassObject-class), 4
- mass<- (AbstractMassObject-class), 4
- mass<- ,AbstractMassObject,numeric-method
 - (AbstractMassObject-class), 4
- MassPeaks, 2–6, 8, 11–13, 18, 19, 23–26, 28, 31, 34, 37
- MassPeaks (MassPeaks-class), 26
- MassPeaks-class, 26
- MassSpectrum, 2–5, 9–11, 15, 16, 23, 24, 26–29, 32–34, 36, 37
- MassSpectrum (MassSpectrum-class), 26
- MassSpectrum-class, 26
- matrix, 21
- mean, 28
- median, 28
- mergeMassPeaks (mergeMassSpectra), 28
- metaData (AbstractMassObject-class), 4
- metaData, AbstractMassObject-method
 - (AbstractMassObject-class), 4
- metaData<- (AbstractMassObject-class), 4
- metaData<- ,AbstractMassObject-method
 - (AbstractMassObject-class), 4

- par, 25, 29, 30
- plot, 30
- plot, AbstractMassObject,missing-method
 - (plot-methods), 29
- plot, MassSpectrum,missing-method
 - (plot-methods), 29
- plot-methods, 29
- plot.default, 30
- points, 5
- points, AbstractMassObject-method
 - (AbstractMassObject-class), 4

- referencePeaks, 12, 13, 31
- removeBaseline
 - (removeBaseline-methods), 32
- removeBaseline,list-method
 - (removeBaseline-methods), 32
- removeBaseline,MassSpectrum-method
 - (removeBaseline-methods), 32
- removeBaseline-methods, 32
- removeEmptyMassObjects
 - (findEmptyMassObjects), 20

- sqrt, 35
- standardizeTotalIonCurrent, 34
- standardizeTotalIonCurrent
 - (standardizeTotalIonCurrent-methods), 34
- standardizeTotalIonCurrent,list-method
 - (standardizeTotalIonCurrent-methods), 34
- standardizeTotalIonCurrent-methods, 34
- sum, 28

- text, 25
- title, 30
- totalIonCurrent
 - (AbstractMassObject-class), 4
- totalIonCurrent,AbstractMassObject-method
 - (AbstractMassObject-class), 4
- totalIonCurrent<-
 - (AbstractMassObject-class), 4

`totalIonCurrent<-`, `AbstractMassObject`, numeric-method
 (`AbstractMassObject`-class), 4

`transformIntensity`, 35

`transformIntensity`
 (`transformIntensity`-methods),
 35

`transformIntensity`, `AbstractMassObject`-method
 (`transformIntensity`-methods),
 35

`transformIntensity`, list-method
 (`transformIntensity`-methods),
 35

`transformIntensity`-methods, 35

TRUE, 22, 23

vector, 20

`warpMassPeaks`, 13

`warpMassPeaks` (`warpMassSpectra`), 36

`warpMassSpectra`, 13, 36