

# Package ‘CAMERA’

October 18, 2022

**Version** 1.52.0

**Date** 2022-02-07

**Title** Collection of annotation related methods for mass spectrometry data

**Author** Carsten Kuhl, Ralf Tautenhahn, Hendrik Treutler, Steffen Neumann {ck-uhllhtreutle|sneumann}@ipb-halle.de, rtautenh@scripps.edu

**Maintainer** Steffen Neumann <sneumann@ipb-halle.de>

**Depends** R (>= 2.1.0), methods, Biobase, xcms (>= 1.13.5)

**Imports** methods, xcms, RBGL, graph, graphics, grDevices, stats, utils, Hmisc, igraph

**Suggests** faahKO, RUnit, BiocGenerics

**Enhances** Rmpi, snow

**Description** Annotation of peaklists generated by xcms, rule based annotation of isotopes and adducts, isotope validation, EIC correlation based tagging of unknown adducts and fragments

**License** GPL (>= 2)

**ByteCompile** TRUE

**URL** <http://msbi.ipb-halle.de/msbi/CAMERA/>

**BugReports** <https://github.com/sneumann/CAMERA/issues/new>

**biocViews** ImmunoOncology, MassSpectrometry, Metabolomics

**RoxygenNote** 5.0.1

**git\_url** <https://git.bioconductor.org/packages/CAMERA>

**git\_branch** RELEASE\_3\_15

**git\_last\_commit** 63642b8

**git\_last\_commit\_date** 2022-04-26

**Date/Publication** 2022-10-18

**R topics documented:**

annotate-methods . . . . .	3
annotateDiffreport . . . . .	4
calcCaS-methods . . . . .	6
calcCiS-methods . . . . .	7
calcIsotopes-methods . . . . .	8
calcPC-methods . . . . .	9
calcPC.hcs . . . . .	10
calcPC.lpc . . . . .	11
cleanParallel . . . . .	12
combinexsAnnos . . . . .	12
compoundLibraries . . . . .	13
compoundQuantiles . . . . .	14
compoundQuantiles-class . . . . .	15
findAdducts-methods . . . . .	16
findIsotopes . . . . .	17
findIsotopesWithValidation . . . . .	18
findKendrickMasses . . . . .	19
findNeutralLoss . . . . .	20
findNeutralLossSpecs . . . . .	21
getAllPeakEICs . . . . .	22
getAtomCount,compoundQuantiles-method . . . . .	23
getIsotopeCluster . . . . .	24
getIsotopeProportion,compoundQuantiles-method . . . . .	25
getPeaklist . . . . .	26
getpspectra . . . . .	27
getReducedPeaklist . . . . .	28
groupCorr . . . . .	29
groupDen . . . . .	31
groupFWHM . . . . .	32
massWindowSizes . . . . .	33
mm14 . . . . .	34
plotEICs-methods . . . . .	35
plotPsSpectrum-methods . . . . .	36
psDist-methods . . . . .	37
pspec2metfrag . . . . .	38
ruleSet . . . . .	39
xsAnnotate . . . . .	40
xsAnnotate-class . . . . .	41

### Description

Wrapper skript for automatic annotation of isotope peaks, adducts and fragments for a (grouped) xcmsSet xs. The function returns an xsAnnotate object.

### Usage

```
annotate(object, sample=NA, nSlaves=1, sigma=6, perfw hm=0.6,
  cor_eic_th=0.75, graphMethod="hcs", pval=0.05, calcCiS=TRUE,
  calcIso=FALSE, calcCaS=FALSE, maxcharge=3, maxiso=4, minfrac=0.5,
  ppm=5, mzabs=0.015, quick=FALSE, psg_list=NULL, rules=NULL,
  polarity="positive", multiplier=3, max_peaks=100 ,intval="into")
```

### Arguments

object	xcmsSet with peak group assignments
sample	xsAnnotate: Sample selection for grouped xcmsSet, see <a href="#">xsAnnotate-class</a>
nSlaves	xsAnnotate: Use parallel CAMERA mode, require Rmpi
sigma	groupFWHM: multiplier of the standard deviation
perfw hm	groupFWHM: percentage of FWHM width
cor_eic_th	groupCorr: correlation threshold (0..1)
graphMethod	groupCorr: Method selection for grouping peaks after correlation analysis into pseudospectra
pval	groupCorr: significant correlation threshold
calcCiS	groupCorr: Use correlation inside samples for peak grouping
calcIso	groupCorr: Use isotopic relationship for peak grouping
calcCaS	groupCorr: Use correlation across samples for peak grouping
maxcharge	findIsotopes: max. ion charge
maxiso	findIsotopes: max. number of expected isotopes
minfrac	findIsotopes: The percentage number of samples, which must satisfy the C12/C13 rule for isotope annotation
ppm	General ppm error
mzabs	General absolut error in m/z
quick	Use only groupFWHM and findIsotopes
psg_list	Calculation will only be done for the selected groups
rules	findAdducts: User defined ruleset
polarity	findAdducts: Which polarity mode was used for measuring of the ms sample
multiplier	findAdducts: If no ruleset is provided, calculate ruleset with max. number n of [nM+x] clusterions
max_peaks	How much peaks will be calculated in every thread using the parallel mode
intval	General used intensity value (into, maxo, intb)

**Details**

Batch script for annotation of an (grouped) `xcmsSet` `xs`. Generates an `xsAnnotate` object by calling all involved functions for the annotation step. Function list: 1: `groupFWHM()`, 2: `findIsotopes()`, 3: `groupCorr()`, 4: `findAdducts()` Return the `xsAnnotate` object, which inherits all annotations. For more information about the parameters see the specific function manpages.

**Value**

`annotate` returns an `xsAnnotate` object. For more information about the `xsAnnotate` object see [xsAnnotate-class](#).

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- annotate(xs)
```

---

annotateDiffreport      *Automatic deconvolution/annotation of LC/ESI-MS data*

---

**Description**

Wrapper function for the `xcms` `diffreport` and the `annotate` function. Returns a `diffreport` within the annotation results.

**Usage**

```
annotateDiffreport(object, sample=NA, nSlaves=1, sigma=6, perfwhm=0.6,
  cor_eic_th=0.75, cor_exp_th = 0.75, graphMethod="hcs", pval=0.05, calcCiS=TRUE,
  calcIso=FALSE, calcCaS=FALSE, maxcharge=3, maxiso=4, minfrac=0.5,
  ppm=5, mzabs=0.015, quick=FALSE, psg_list=NULL, rules=NULL,
  polarity="positive", multiplier=3, max_peaks=100, intval="into",
  pval_th = NULL, fc_th = NULL, sortpval=TRUE, ...)
```

**Arguments**

<code>object</code>	<code>xcmsSet</code> with peak group assignments
<code>sample</code>	<code>xsAnnotate</code> : Sample selection for grouped <code>xcmsSet</code> , see <a href="#">xsAnnotate-class</a>
<code>nSlaves</code>	<code>xsAnnotate</code> : Use parallel CAMERA mode, require <code>Rmpi</code>
<code>sigma</code>	<code>groupFWHM</code> : multiplier of the standard deviation
<code>perfwhm</code>	<code>groupFWHM</code> : percentage of FWHM width

cor_eic_th	groupCorr: Correlation threshold for EIC correlation (0..1)
cor_exp_th	groupCorr: Threshold for intensity correlations across samples (0..1)
graphMethod	groupCorr: Method selection for grouping peaks after correlation analysis into pseudospectra
pval	groupCorr: significant correlation threshold
calcCiS	groupCorr: Use correlation inside samples for peak grouping
calcIso	groupCorr: Use isotopic relationship for peak grouping
calcCaS	groupCorr: Use correlation across samples for peak grouping
maxcharge	findIsotopes: max. ion charge
maxiso	findIsotopes: max. number of expected isotopes
minfrac	findIsotopes: The percentage number of samples, which must satisfy the C12/C13 rule for isotope annotation
ppm	General ppm error
mzabs	General absolut error in m/z
quick	Use only groupFWHM and findIsotopes
psg_list	Calculation will only be done for the selected groups
rules	findAdducts: User defined ruleset
polarity	findAdducts: Which polarity mode was used for measuring of the ms sample
multiplier	findAdducts: If no ruleset is provided, calculate ruleset with max. number n of [nM+x] clusterions
max_peaks	How much peaks will be calculated in every thread using the parallel mode
intval	General used intensity value (into, maxo, intb)
pval_th	pval threshold. Creates a new psg_list. A pseudospectra is selected if it contains peaks, with pval < pval_th
fc_th	Same as pval. Select those groups with contains peaks with fold-change > fc_th. Pval_th and fc_th can be combined
sortpval	Sort diffreport after pvalues
...	Diffreport parameters see <a href="#">diffreport</a>

### Details

Batch script wrapper for combining the annotation and the diffreport for a (grouped) xcmsSet xs. Function list: 1: diffreport(), 2: groupFWHM(), 3: findIsotopes(), 4: groupCorr(), 5: findAdducts() For a speedup calculation users can create a quick run, with quick = TRUE to preselect pseudospectra of interest. The indices of those pseudospectra are set with psg\_list in a second run. On the other hand, a automatic selection with pval\_th and/or fc\_th can be performed. Returns the normal xcms diffreport table, with the additional CAMERA slots

### Value

annotateDiffreport returns an diffreport, see [diffreport](#), within additional columns containing the annotation results.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
#Multiple sample
library(CAMERA)
library(faahKO)
xs.grp      <- group(faahko)
xs.fill     <- fillPeaks(xs.grp)

#fast preselection
# diffreport <- annotateDiffreport(xs.fill,quick=TRUE)
# index <- c(1,18,35,45,56) #Make only for those grps a adduct annotation
# diffreport2 <- annotateDiffreport(xs.fill,psg_list=index,metlin = TRUE)

#automatic selection for groups with peaks p-val < 0.05 and fold-change > 3
# diffreport <- annotateDiffreport(xs.fill,pval_th=0.05,fc=3)
```

---

calcCaS-methods

*EIC correlation grouping of LC/ESI-MS data*

---

**Description**

Calculate the correlation across samples. Filtering correlation with specific parameters and returns a correlation matrix.

**Usage**

```
calcCaS(object,corval=0.75, pval=0.05, intval="into")
```

**Arguments**

object	The xsAnnotate object
corval	Correlation threshold for positive hits
pval	P-Value threshold for significance level of correlation
intval	Selection of the intensity values that should be used in the correlation analysis. Can be into, maxo or intb.

**Details**

Calculate pearson correlation between the peak intensities over all samples. Afterwards use cor.test for returning only significant correlation. Returns only those correlation, which are above both threshold. Set corval and pval to 0 to get the unfiltered correlation matrix. If the object is pregrouped with groupFWHM, then the correlation is only calculated between peaks within a pseudospectrum. Otherwise between all peaks.

**Value**

A matrix with 4 columns:

x	peak index according to peaktable
y	peak index according to peaktable
cor	correlation value between peak x and peak y
ps	pseudospektrum index for both peaks

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**See Also**

[calcCiS groupCorr xsAnnotate-class](#)

**Examples**

```
library(CAMERA)
#Multiple sample
library(faahKO)
xs.grp      <- group(faahko)
#create xsAnnotate object
xsa        <- xsAnnotate(xs.grp)
#generate pseudospectra
xsa.group  <- groupFWHM(xsa)
#calculate correlation
correlationMatrix <- calcCaS(xsa.group)
```

---

calcCiS-methods

*Calculate peak distance matrix after EIC correlation*

---

**Description**

Processing an xsAnnotate object and correlates peak EIC curves from one pseudospectrum, using a precalculated EIC matrix ([getAllPeakEICs](#)). It return a weighted edge list as distance matrix between peaks according to the correlation analysis. The edge value is the pearson correlation coefficient. The list can be used as input for [calcPC](#).

**Usage**

```
calcCiS(object, EIC=EIC, corval=0.75, pval=0.05, psg_list=NULL)
```

**Arguments**

object	The xsAnnotate object
EIC	EIC Matrix
corval	Correlation threshold for the EIC correlation
pval	pvalue for testing correlation of significance
psg_list	Vector of pseudospectra indices. The correlation analysis will be only done for those groups

**Details**

The algorithm correlates the EIC of a every peak with all others, to find the peaks that belong to one substance. LC/MS data should grouped with groupFWHM first. This step reduce the runtime a lot and increased the number of correct classifications. Only correlation with a higher value than the correlation threshold and significant p-values will be returned.

**Value**

A matrix with 4 columns:

x	peak index
y	peak index
cor	correlation value
ps	pseudospectrum index, which contains x and y

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**See Also**

[calcCaS](#) [groupCorr](#) [getAllPeakEICs](#) [xsAnnotate-class](#)

---

calcIsotopes-methods    *Calculate isotope distance matrix from xsAnnotate object*

---

**Description**

Processing an xsAnnotate object with annotated isotopes ([findIsotopes](#)). It return a weighted edge list as distance matrix between peaks according to the isotope annotation. The edge value for recognized isotopes is 1 for all cases. The list can be used as input for [calcPC](#).

**Arguments**

object	xsAnnotate object
--------	-------------------



**Value**

A matrix with 4 columns:

x	peak index
y	peak index
cor	edge value, always 1
ps	pseudospectrum index, which contains x and y

**Methods**

```
object = "xsAnnotate" calcIsotopes(object)
```

**Author(s)**

Carsten Kuhl, <ckuhl@ipb-halle.de>

**See Also**

[calcPC xsAnnotate-class](#)

---

calcPC-methods

*Peakclustering into pseudospectra according to a distance matrix*

---

**Description**

A number of clustering methods exist in CAMERA. calcPC is the generic method.

**Usage**

```
calcPC(object, method, ...)
```

**Arguments**

object	<a href="#">xsAnnotate-class</a> object
method	Method to use for clustering. See details.
...	Optional arguments to be passed along

**Details**

This algorithms cluster peaks from a xsAnnotate object into pseudospectra according to a provided distance matrix. Therefore all peaks are transformend into a graph, with peaks as nodes and the value from the distance matrix as edges. Afterwards a graph separation algorithm is applied, which searches in the graph for clusters. See the manpages of the specific clustering algorithms for more information.

If the xsAnnotate is pregrouped, for example [groupFWHM](#), only the already existing groups will be further processed.

The different algorithms that can be used by specifying them with the method argument. For example to use the highly connected subgraphs approach by E. Hartuv, R. Shamir, (1999), one would use: `calcPC(object, method="hcs")`. This is also the default, see [calcPC.hcs](#).

Further arguments given by `...` are passed through to the function implementing the method, which are most likely `ajc`. The parameter `ajc` is the peak distance matrix.

`getOption("BioC")$CAMERA$findPeaks.methods` returns a character vector of *nicknames* for the algorithms available.

The function returns a `xsAnnotate` object with grouping information, as list of peak indices. They are stored as `object@pspectra`.

### See Also

[calcPC.lpc](#) [calcPC.hcs](#) [xsAnnotate-class](#)

---

calcPC.hcs

*Peakclustering into pseudospectra with the highly connected subgraphs approach*

---

### Description

Cluster peaks from an `xsAnnotate` object into pseudospectra

### Arguments

<code>object</code>	<code>xsAnnotate</code> object
<code>ajc</code>	Weighted symbolic edge list as four column matrix ("x","y","cor","ps"). Columns x,y are peak indices, cor the edge value and ps the pseudospectrum index, where both peaks occur.
<code>psg_list</code>	additional vector ps pseudospectra indices, which are used in the clustering. If set to NULL all pseudospectra will be processed.

### Details

In some cases, is the peak grouping after retentiontime with [groupFWHM](#) not enough to separate co-elution compounds. Therefore [groupCorr](#) use additional correlation analysis to achieve a separation. `calcPC` is part of this approach, which takes the calculated weighted edge list and performs the graph clustering. It returns an `xsAnnotate` object with further separated pseudospectra.

### Methods

**object = "xsAnnotate"** `calcPC.hcs(object, ajc=NULL, psg_list=NULL)`

### Author(s)

Carsten Kuhl, <ckuhl@ipb-halle.de>

**See Also**

[calcPC](#) [groupCorr](#) [highlyConnSG](#) [xsAnnotate-class](#)

---

calcPC.lpc	<i>Peakclustering into pseudospectra with the label-propagation-community algorithm</i>
------------	-----------------------------------------------------------------------------------------

---

**Description**

Cluster peaks from an `xsAnnotate` object into pseudospectra

**Arguments**

<code>object</code>	<code>xsAnnotate</code> object
<code>ajc</code>	Weighted symbolic edge list as four column matrix (" <code>x</code> ", " <code>y</code> ", " <code>cor</code> ", " <code>ps</code> "). Columns <code>x</code> , <code>y</code> are peak indices, <code>cor</code> the edge value and <code>ps</code> the pseudospectrum index, where both peaks occur.
<code>psg_list</code>	additional vector <code>ps</code> pseudospectra indices, which are used in the clustering. If set to <code>NULL</code> all pseudospectra will be processed.

**Details**

In some cases, is the peak grouping after retentiontime with [groupFWHM](#) not enough to separate co-elution compounds. Therefore [groupCorr](#) use additional correlation analysis to achieve a separation. `calcPC` is part of this approach, which takes the calculated weighted edge list and performs the graph clustering. It returns an `xsAnnotate` object with further separated pseudospectra.

**Methods**

```
object = "xsAnnotate" calcPC.lpc(object, ajc=NULL, psg_list=NULL)
```

**Author(s)**

Carsten Kuhl, <ckuhl@ipb-halle.de>

**See Also**

[calcPC](#) [groupCorr](#) [xsAnnotate-class](#) [label.propagation.community](#)

---

cleanParallel	<i>Cleans up with spawned slave processes after use</i>
---------------	---------------------------------------------------------

---

**Description**

The spawned slaves processes, which are created within the parallel mode, are closed explicit.

**Usage**

```
cleanParallel(object)
```

**Arguments**

object           xsAnnotate object

**Details**

The function needs a xsAnnotate object after groupCorr or groupFWHM. The resulting object is a artificial xcmsSet, where the peaks with the specific neutral loss are stored in xcmsSet@peaks.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
## Not run:  library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs  <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an  <- xsAnnotate(xs, polarity="positive", nSlaves=2)
an  <- groupFWHM(an)
an  <- findAdducts(an)
cleanParallel(an)

## End(Not run)
```

---

combinexsAnnos	<i>Check CAMERA ion species annotation due to matching with opposite ion mode</i>
----------------	-----------------------------------------------------------------------------------

---

**Description**

This function check annoations of ion species with the help of a sample from opposite ion mode. As first step it searches for pseudospectra from the positive and the negative sample within a retention time window. For every result the m/z differences between both samples are matched against specific rules, which are combinations from pos. and neg. ion species. As example M+H and M-H with a m/z difference of 2.014552. If two ions matches such a difference, the ion annotations are changed (previous annotation is wrong), confirmed or added. Returns the peaklist from one ion mode with recalculated annotations.

**Usage**

```
combinexsAnnos(xsa.pos, xsa.neg, pos=TRUE, tol=2, ruleset=NULL)
```

**Arguments**

xsa.pos	xsAnnotate object with positive ion mode
xsa.neg	xsAnnotate object with neagive ion mode
pos	If TRUE the peaklist from the positive mode is returned, if FALSE the negative
tol	Retention time window in seconds
ruleset	Matrix of matching rules, see example

**Details**

Both xsAnnotate object should be full processed (grouping and annotation). Without previous annotation the resulting peaklist only includes annotation with matches peaks from both mode according to the rule(s). With ruleset=NULL the function only looks for M+H/M-H pairs. The ruleset is a two column matrix with includes rule indices from the rule table of both xsAnnotate objects. `ruleset <- cbind(1,1)` would create the M+H/M-H rule, since the first rule of `xsa.pos@ruleset` and `xsa.neg@ruleset` is M+H respectively M-H. Only rules with identical charge can be combined!

**Value**

Returns a (normal) CAMERA peaklist with a additional column `neg. Mode` or `pos. Mode`, where matching peaks from the opposite mode are noted.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
## Not run:
#Searches for M+H/M-H combinations within a retention time window of 2 seconds
peaklist.pos <- combinexsAnnos(xsa.pos, xsa.neg, tol=2)

## End(Not run)
```

---

compoundLibraries      *The supported compound databases*

---

**Description**

Returns a set of supported compound databases

**Usage**

```
compoundLibraries()
```

**Value**

Vector of supported compound databases

**Author(s)**

Hendrik Treutler

**Examples**

```
compoundLibraries()
```

---

compoundQuantiles      *compoundQuantiles constructor*

---

**Description**

constructor of class compoundQuantiles

**Usage**

```
compoundQuantiles(compoundLibrary = "kegg", massWindowSize = 50)
```

**Arguments**

compoundLibrary

the database; see compoundLibraries() for a list of supported databases

massWindowSize the mass window size for grouping compounds; see massWindowSizes(compoundLibrary = "kegg") for a list of supported databases for e.g. the database kegg

**Value**

the compoundQuantiles object

**Author(s)**

Hendrik Treutler

**Examples**

```
cpObj <- compoundQuantiles()
```

---

 compoundQuantiles-class

*Class compoundQuantiles encapsulates compound statistics from different databases.*

---

## Description

The user is able to get the expected number of atoms of element e (C, N, ...) for a compound of mass m for a q-quantile. I.e. `getAtomCount(object = compoundQuantiles(), element = e, mass = m, quantile = q)` returns the number of atoms of element e in a compound of mass m in the lowest-(q\*100) (sorted ascending by the possible number of atoms of element e for compounds of such mass).

The user is able to get the expected proportion between the intensities of two isotope peaks for a compound of mass m for a q-quantile. I.e. `getIsotopeProportion(object = compoundQuantiles(), isotope1 = i1, isotope2 = i2, mass = m, quantile = q)` returns the isotope proportion  $i1 / i2$  for a compound of mass m in the lowest-(q\*100) (sorted ascending by the possible isotope proportions for compounds of such mass).

## Objects from the Class

Objects can be created with the `compoundQuantiles` constructor.

## Slots

`compoundLibrary`: The compound library to rely on (kegg, chebi, ...)

`massWindowSize`: The mass window size of the compound statistics (25, 100, ...)

`minCompoundMass`: Minimum compound mass for which there are statistics

`maxCompoundMass`: Maximum compound mass for which there are statistics

`numberOfMassWindows`: Number of mass windows

`numberOfIsotopes`: Number of isotopes for which there are isotope ratio quantiles

`isotopeSet`: The set of isotopes for which there are isotope ratio quantiles

`elementSet`: The set of elements for which there are element count statistics

`quantileSet`: The set of quantiles for which there are isotope ratio statistics

`eleCounters_e_q_mw`: Three dimensional array containing the element count statistics (element, quantile, mass window index)

`proportions_i_q_mw`: Three dimensional array containing the isotope ratio quantiles relative to the monoisotopic peak (isotope index, quantile, mass window index)

## Methods

**`getAtomCount`** signature(object = "xsAnnotate"): returns the number of atoms of the specified element for the given quantile and mass window index

**`getIsotopeProportion, compoundQuantiles-method`** signature(object = "xsAnnotate"): returns the isotope ratio of the specified isotope for the given quantile and mass window index relative to the monoisotopic peak

**Note**

No notes yet.

**Author(s)**

Hendrik Treutler, <hendrik.treutler@ipb-halle.de>

**See Also**

[compoundQuantiles](#) [getAtomCount](#) [getIsotopeProportion](#)

---

findAdducts-methods     *Calculate Adducts and Annotate LC/ESI-MS Spectra*

---

**Description**

Annotate adducts (and fragments) for a xsAnnotate object. Returns a xsAnnotate object with annotated pseudospectra.

**Usage**

```
findAdducts(object, ppm=5, mzabs=0.015, multiplier=3,  
polarity=NULL, rules=NULL, max_peaks=100, psg_list=NULL, intval="maxo")
```

**Arguments**

object	the xsAnnotate object
ppm	ppm error for the search
mzabs	allowed variance for the search
multiplier	highest number(n) of allowed clusterion [nM+ion]
polarity	Which polarity mode was used for measuring of the ms sample
rules	personal ruleset or with NULL standard ruleset will be calculated
max_peaks	If run in parralel mode, this number defines how much peaks will be calculated in every thread
psg_list	Vector of pseudospectra indices. The correlation analysis will be only done for those groups
intval	choose intensity values. Allowed values are into, maxo, intb

**Details**

Adducts (and fragments) are annotated for a xsAnnotate object. For every pseudospectra group, generated bei groupFWHM and groupCorr, all possible Adducts are calculated and mapped to the peaks. If at least two adducts match, a possible molecule-mass for the group can be calculated. After the annotation every masshypothese is checked against the charge of the calculated isotopes. It is recommend to call findIsotopes() before the annotation step.



**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an) # optional but recommended.
#an <- groupCorr(an) # optional but very recommended step
an <- findAdducts(an,polarity="positive")
peaklist <- getPeaklist(an) # get the annotated peak list
```

---

findIsotopes

*Deconvolute/Annotate LC/ESI-MS data*

---

**Description**

Annotate isotope peaks for a xsAnnotate object. Returns a xsAnnotate object with annotated isotopes.

**Usage**

```
findIsotopes(object, maxcharge=3, maxiso=4, ppm=5, mzabs=0.01, intval=c("maxo","into","intb"), minfrac
```

**Arguments**

object	the xsAnnotate object
maxcharge	max. number of the isotope charge
maxiso	max. number of the isotope peaks
ppm	ppm error for the search
mzabs	allowed variance for the search
intval	choose intensity values for C12/C13 check. Allowed values are into, maxo, intb
minfrac	in case of multiple samples, percentaged value of samples, which have to contain the correct C12/C13 ratio and are not NA
isotopeMatrix	four column m/z-diff and ratio Matrix, for matching isotopic peaks.
filter	Should C12/C13 filter be applied

**Details**

Isotope peaks are annotated for a `xsAnnotate` object according to given rules (`maxcharge`, `maxiso`). The algorithm benefits from a earlier grouping of the data, with `groupFWHM`. Generates a list of all possible isotopes, which is stored in `object@isotopes`. Those isotope information will be used in the `groupCorr` function. The intensity of the C13 isotope peak is checked against the C12 of proper ratio. In the case of multiple sample, all samples will be tested. `Minfrac` describe the minimal percentage of samples, which must passed the test. If peaks are NA, then this sample is skipped and the ratio is  $(\text{found correct C12/C13 ratio}) / (\text{samples containing C12 and C13 peak})$ .

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
```

---

findIsotopesWithValidation

*Deconvolute/Annotate LC/ESI-MS data*

---

**Description**

Annotate validated isotope clusters for a `xsAnnotate` object. Returns a `xsAnnotate` object with annotated isotopes. Validation of isotope clusters is based on statistics of the KEGG database implemented in S4 class object `compoundQuantiles`.

**Usage**

```
findIsotopesWithValidation(object, maxcharge=3, ppm=5, mzabs=0.01, intval=c("maxo", "into", "intb"), validateIsotopePatterns=TRUE, database="KEGG")
```

**Arguments**

<code>object</code>	the <code>xsAnnotate</code> object
<code>maxcharge</code>	max. number of the isotope charge
<code>ppm</code>	ppm error for the search
<code>mzabs</code>	allowed variance for the search
<code>intval</code>	choose intensity values for C12/C13 check. Allowed values are <code>into</code> , <code>maxo</code> , <code>intb</code>
<code>validateIsotopePatterns</code>	logical, if TRUE putative isotope clusters are validated based on KEGG database statistics.
<code>database</code>	the database which is the basis for isotope cluster validation. One of <code>compoundLibraries()</code> .

## Details

Isotope peaks are annotated for a `xsAnnotate` object according to given rules (`maxcharge`, `maxiso`). The algorithm benefits from a earlier grouping of the data, with `groupFWHM`. Generates a list of all possible isotopes, which is stored in `object@isotopes`. Those isotope information will be used in the `groupCorr` function. The ratios between isotope peaks are checked against the mass-specific 99% confidence interval based on statistics of the KEGG database.

## Author(s)

Hendrik Treutler <hendrik.treutler@ipb-halle.de>

## References

Hendrik Treutler and Steffen Neumann. "Prediction, detection, and validation of isotope clusters in mass spectrometry data". Submitted to *Metabolites* 2016, Special Issue "Bioinformatics and Data Analysis".

## See Also

[findIsotopes](#)

## Examples

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopesWithValidation(an)
```

---

findKendrickMasses      *Find specific mass defects using Kendrick mass scales*

---

## Description

Todo

## Usage

```
findKendrickMasses(object, masses=c(14, 14.01565),
maxHomologue=4, error=0.002, time=60, intval="maxo",
plot=FALSE)
```

**Arguments**

object	xsAnnotate object
masses	nominal mass and exact mass
error	allowed mass difference in Da for matching Kendrick mass defect
maxHomologue	max number of homologue
time	allowed retention time difference between homologues
intval	intensity value (allowed values: maxo,into or intb)
plot	plot hits

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
library(faahKO)
xs <- group(faahko)

#With specific selected sample
xsa <- xsAnnotate(xs)
#Screen for substance with CH2 differences
findKendrickMasses(xsa, masses=c(14, 14.01565), plot=TRUE)
```

---

findNeutralLoss      *Find pseudospectra that contains a specific neutral loss*

---

**Description**

The method searches in every pseudospectra for a distance between two ions matching a provided mass difference. It returns a xcmsSet object containing the matching peaks.

**Usage**

```
findNeutralLoss(object, mzdiff=NULL, mzabs=0, mzppm=10)
```

**Arguments**

object	xsAnnotate object
mzdiff	neutral loss in Dalton
mzabs	absolut allowed mass difference
mzppm	relative allowed mass difference

**Details**

The function needs a `xsAnnotate` object after `groupCorr` or `groupFWHM`. The resulting object is a artificial `xcmsSet`, where the peaks with the specific neutral loss are stored in `xcmsSet@peaks`.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
#Searches for Peaks with water loss
xs.pseudo <- findNeutralLoss(an,mzdiff=18.01,mzabs=0.01)
xs.pseudo@peaks #show Hits
```

---

findNeutralLossSpecs *Find pseudospectra that contains a specific neutral loss*

---

**Description**

The method searches in every pseudospectra for a distance between two ions matching a provided mass difference. It returns a boolean vector with the length equals to the number of pseudospectra, where a hit is marked with TRUE.

**Usage**

```
findNeutralLossSpecs(object, mzdiff=NULL, mzabs=0, mzppm=10)
```

**Arguments**

<code>object</code>	<code>xsAnnotate</code> object
<code>mzdiff</code>	neutral loss in Dalton
<code>mzabs</code>	absolut allowed mass difference
<code>mzppm</code>	relative allowed mass difference

**Details**

The function needs a `xsAnnotate` object after `groupCorr` or `groupFWHM`.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
#Searches for Pseudospectra with water loss
hits <- findNeutralLossSpecs(an, mzdiff=18.01, mzabs=0.01)
```

---

getAllPeakEICs	<i>Generate EIC information from raw data</i>
----------------	-----------------------------------------------

---

**Description**

Generate EIC data out of the raw data, according to the peak peaker information.

**Usage**

```
getAllPeakEICs(object, index)
```

**Arguments**

object	The xsAnnotate object
index	Sample index vector, with the same length as the number of peaks. Encoding from with sample the peak should be extracted. If all peaks should be generated from the same sample set index = rep(sample index, peak count)

**Details**

The function extract from the raw data the EIC curves. Therefore all .netcdf, .mzML etc. files must be accessible. It returns a list with two item.

**Value**

A list with items:

EIC	EIC Matrix with rows = number of peaks and columns = maxscans. It contains mostly NA values and only in that part, where a peak had been found, the intensity information.
scantimes	Scantimes of each sample

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**See Also**

[xsAnnotate-class](#)

**Examples**

```
library(CAMERA)
#Multiple sample
library(faahKO)
xs.grp      <- group(faahko)

#create xsAnnotate object
xsa        <- xsAnnotate(xs.grp)
#generate pseudospectra
xsa.group  <- groupFWHM(xsa)

#calculate correlation
tmp <- getAllPeakEICs(xsa.group,index=rep(1,nrow(xsa.group@groupInfo)))
#extract EIC matrix
EIC.matrix <- tmp$EIC;
```

---

getAtomCount,compoundQuantiles-method

*The number of atoms of the given element*

---

**Description**

Returns the number of atoms the specified element in a compound of the specified mass for the specified quantile level

**Usage**

```
## S4 method for signature 'compoundQuantiles'
getAtomCount(object, element, mass, quantile)
```

**Arguments**

object	A compoundQuantiles object
element	The element of interest specified by element symbol
mass	The mass of the compound specified in atomic units (=dalton)
quantile	The quantile level for the number of atoms

**Value**

The number of atoms

**Author(s)**

Hendrik Treutler

## Examples

```
cpObj <- compoundQuantiles()

compoundMass <- 503
quantileLow <- 0.05
quantileHigh <- 0.95
element <- "C"
countLow <- getAtomCount(object = cpObj, element = element, mass = compoundMass, quantile = quantileLow)
countHigh <- getAtomCount(object = cpObj, element = element, mass = compoundMass, quantile = quantileHigh)

print(paste("The ", (quantileHigh - quantileLow) * 100, "% confidence interval for the number of atoms of element ",
```

---

getIsotopeCluster      *Retrieve the annotated isotopes*

---

## Description

Extract all annotated isotope cluster. Returns a list with one element per cluster. A element contains the charge of the molecule and a peakmatrix with m/z and intensity value.

## Usage

```
getIsotopeCluster(object, number=NULL, value="maxo", sampleIndex=NULL)
```

## Arguments

object	xsAnnotate object
number	Set to NULL extract all isotope cluster or to specific chosen ones
value	Which intensity values should be extracted. Allowed values are: maxo, into, intb
sampleIndex	Selection vector with indexes to select from which sample(s) the intensity values should be retrieved. If set to NULL the sample is selected, which has been chosen for the pseudospectra in the grouping step

## Details

This method extract the isotope annotation from a xsAnnotate object. The order of the resulting list is the same as the one in the peaklist, see [getPeaklist](#).

## Author(s)

Carsten Kuhl <ckuhl@ipb-halle.de>



**Examples**

```

#single sample
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
isolist <- getIsotopeCluster(an)
isolist[[10]] #get IsotopeCluster 10

#multiple sample
library(faahK0)
xs <- group(faahko)
xs <- fillPeaks(xs)
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
isolist <- getIsotopeCluster(an)

#Select from multiple samples

isolist <- getIsotopeCluster(an, sampleIndex=c(1,2,5))

##Interaction with Rdisop
## Not run:
library(Rdisop)
isotopes.decomposed <- lapply(isolist,function(x) {
  decomposeIsotopes(x$peaks[,1],x$peaks[,2],z=x$charge);
}) #decomposed isotope cluster, filter steps are recommended

## End(Not run)

```

---

getIsotopeProportion,compoundQuantiles-method

*The proportion of the intensities of two isotope peaks*

---

**Description**

Returns the proportion of the intensities of isotope1 versus isotope2 for a compound of the given mass for the given quantile level

**Usage**

```

## S4 method for signature 'compoundQuantiles'
getIsotopeProportion(object, isotope1, isotope2,
  mass, quantile)

```

**Arguments**

object	A compoundQuantiles object
isotope1	The dividend isotope ranging from 0 (the monoisotopic peak) to 5
isotope2	The divisor isotope ranging from 0 (the monoisotopic peak) to 5
mass	The mass of the compound specified in atomic units (=dalton)
quantile	The quantile level for the isotope proportion

**Value**

The isotope proportion

**Author(s)**

Hendrik Treutler

**Examples**

```
cpObj <- compoundQuantiles(compoundLibrary = "kegg")
```

```
compoundMass <- 503
isotope1 <- 0
isotope2 <- 1
quantileLow <- 0.05
quantileHigh <- 0.95
```

```
propLow <- getIsotopeProportion(object = cpObj, isotope1 = isotope1, isotope2 = isotope2, mass = compoundMass, quantile = quantileLow)
propHigh <- getIsotopeProportion(object = cpObj, isotope1 = isotope1, isotope2 = isotope2, mass = compoundMass, quantile = quantileHigh)
print(paste("The ", (quantileHigh - quantileLow) * 100, "% confidence interval for the proportion of isotopes ", isotope1, " and ", isotope2, " is: ", propHigh - propLow))
```

---

getPeaklist

*Generate the annotated peaklist*

---

**Description**

Extract all information from an xsAnnotate object. Returns a peaklist with annotated peaks.

**Usage**

```
getPeaklist(object, intval="into")
```

**Arguments**

object	xsAnnotate object
intval	Choose intensity values. Allowed values are into, maxo, intb, intf, maxf, area, depending on the feature detection algorithm used.

## Details

This function extract the peaktable from an `xsAnnotate` object, containing three additional columns (isotopes, adducts, pseudospectrum) with represents the annotation results. For a grouped `xcmsSet` it returns the grouped peaktable.

## Author(s)

Carsten Kuhl <ckuhl@ipb-halle.de>

## Examples

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
an <- findAdducts(an,polarity="positive")
peaklist <- getPeaklist(an)
```

---

getpspectra

*Retrieve a peaklist of one or more pseudospectra*

---

## Description

Extract group(s) from a `xsAnnotate` object. Returns a peaklist as matrix with annotated peaks.

## Usage

```
getpspectra(object, grp)
```

## Arguments

<code>object</code>	<code>xsAnnotate</code> object
<code>grp</code>	index of pseudo-spectra-group

## Details

`xsAnnotate` groups LC/MS Peaklist after there EIC correlation and FWHM. These function extract one or more of these so called "pseudo spectra groups" with include the peaklist with there annotations. The annotation depends on a before called `findAdducts()` ( and `findIsotopes()` ). Important: The indices for the isotopes, are those from the whole peaklist. See `getPeaklist()`.

## Author(s)

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```

library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(c(file), method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
#For one group
peaklist <- getpspectra(an, 1)
#For two groups
peaklist <- getpspectra(an, c(1,2))

```

---

getReducedPeaklist      *Generate reduced peaklist from the annotated peaklist*

---

**Description**

Extract information from an `xsAnnotate` object. Returns a reduced peaklist with annotated peaks. For any putative compound in the `pcgroup`, all found adducts are pooled into one putative compound per group. Thus, the reduced peaklist only contains one annotated adduct per `pcgroup`.

**Usage**

```
getReducedPeaklist(object, method = "median", intval = "into", default.adduct.info = "first", mzrt.ra
```

**Arguments**

<code>object</code>	<code>xsAnnotate</code> object.
<code>method</code>	Choose reduction method. Allowed values are "sum", "median", "maxint", "pca".
<code>intval</code>	Choose intensity values. Allowed values are "into", "maxo", "intb".
<code>default.adduct.info</code>	Choose method to select adduct information. Allowed values are "first", "maxint", "maxpeaks"
<code>mzrt.range</code>	If TRUE, max and min values of mz and rt values of all adducts within a <code>pcgroup</code> are saved (not recommended).
<code>npeaks.sum</code>	If TRUE, the sum of all peaks of all adducts within a <code>pcgroup</code> is saved (not recommended).
<code>cleanup</code>	If TRUE, NA values and negative abundances are being set to zero and constant features (rows) are being removed.

## Details

This function extracts a reduced peaktable from an `xsAnnotate` object. Normally, all adducts are grouped for any putative compounds and saved within the peaklist (see method [getPeaklist](#)). However, for statistical computation it is sometimes better to only work with putative compounds rather than with all of their adducts. Thus, this function pools all adducts for any putative compound into one putative compound per `pcgroup`. There are several methods to choose from how this is being done. Selection methods: "sum": The intensities of adducts are summed for each sample. "median" (default): The median intensities of adducts is calculated for each sample. "maxint": Only the adduct with the highest intensities throughout the samples is returned. "pca": A Principal Component Analysis is being performed for the adducts for the samples. and the PC1 values are taken as intensity information. Select `mz / rt` methods: "first" (default): The `mz` & `rt` information of the first adduct are taken. "maxint": The `mz` & `rt` information of the adduct that has highest intensities are taken. "maxpeaks": The `mz` & `rt` information of the adduct that has the most peaks are taken. In addition, when `mzrt.range` is `TRUE`, the min and max values of all `mz` and `rt` found in a group are stored within `mzmin`, `mzmax` and `rtmin` and `rtmax` (not recommended). In addition, when `npeaks.sum` is `TRUE`, all peaks within a `pcgroup` are summed (not recommended).

## Author(s)

Kristian Peters <kpeters@ipb-halle.de>

## Examples

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)
an <- findIsotopes(an)
an <- findAdducts(an,polarity="positive")
peaklist.reduced <- getReducedPeaklist(an)
```

---

groupCorr

*EIC correlation grouping of LC/ESI-MS data*

---

## Description

Peak grouping after correlation information into pseudospectrum groups for an `xsAnnotate` object. Return an `xsAnnotate` object with grouping information.

## Usage

```
groupCorr(object,cor_eic_th=0.75, pval=0.05, graphMethod="hcs",
  calcIso = FALSE, calcCiS = TRUE, calcCaS = FALSE, psg_list=NULL, xraw=NULL,
  cor_exp_th=0.75, intval="into", ...)
```

**Arguments**

object	The xsAnnotate object
cor_eic_th	Correlation threshold for EIC correlation
pval	p-value threshold for testing correlation of significance
graphMethod	Clustering method for resulting correlation graph. See <a href="#">calcPC</a> for more details.
calcIso	Include isotope detection informationen for graph clustering
calcCiS	Calculate correlation inside samples
calcCaS	Calculate correlation accross samples
psg_list	Vector of pseudospectra indices. The correlation analysis will be only done for those groups
xraw	Optional xcmsRaw object, which should be used for raw data extraction
cor_exp_th	Threshold for intensity correlations across samples
intval	Selection of the intensity values (such as "into") that should be used in the correlation analysis. See <a href="#">getPeaklist</a> for all allowed values.
...	Additional parameter

**Details**

The algorithm calculates different informations for group peaks into so called pseudospectra. This pseudospectra contains peaks, with have a high correlation between each other. So far three different kind of information are available. Correlation of intensities across samples (need more than 3 samples), EIC correlation between peaks inside a sample and additional the informationen about recognized isotope cluster can be included. After calculation of all these informations, they are combined as edge value into a graph object. A following graph clustering algorithm separate the peaks (nodes in the graph) into the pseudospectra.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**See Also**

[calcCiS](#) [calcCaS](#) [calcPC](#) [xsAnnotate-class](#)

**Examples**

```
library(CAMERA)
file      <- system.file('mzML/MM14.mzML', package = "CAMERA");
xs        <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5, 10));
an        <- xsAnnotate(xs);
an.group  <- groupFWHM(an);
an.iso    <- findIsotopes(an.group); #optional step for using isotope information
an.grp.corr <- groupCorr(an.iso, calcIso=TRUE);

#For csv output
# write.csv(file="peaklist_with_isotopes.csv",getPeaklist(an))
```

```

#Multiple sample
library(faahKO)
xs.grp      <- group(faahko)

#With selected sample
xsa        <- xsAnnotate(xs.grp, sample=1)
xsa.group  <- groupFWHM(xsa)
xsa.iso    <- findIsotopes(xsa.group) #optional step
xsa.grp.corr <- groupCorr(xsa.iso, calcIso=TRUE)

#With automatic selection
xsa.auto   <- xsAnnotate(xs.grp)
xsa.grp    <- groupFWHM(xsa.auto)
xsa.iso    <- findIsotopes(xsa.grp) #optional step
index      <- c(1,4) #Only group one and four will be calculate
#We use also correlation across sample
xsa.grp.corr <- groupCorr(xsa.iso, psg_list=index, calcIso=TRUE, calcCaS=TRUE)
#Note: Group 1 and 4 have no subgroups

```

---

groupDen

*Density-Grouping of LC/ESI-MS data*


---

## Description

Group peaks of a `xsAnnotate` object according to peak distributions in chromatographic time into pseudospectra-groups. Works analogous as the `group.density` method of `xcms`. Returns `xsAnnotate` object with pseudospectra informations.

## Usage

```
groupDen(object, bw = 5 , ...)
```

## Arguments

<code>object</code>	the <code>xsAnnotate</code> object
<code>bw</code>	bandwidth (standard deviation or half width at half maximum) of gaussian smoothing kernel to apply to the peak density chromatogram
<code>...</code>	Further Arguments, NYI

## Details

The grouping strongly depends on the `bw` parameter. For an UPLC a good starting point is smaller or around 1.

## Value

Returns a grouped `xsAnnotate` object.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Examples**

```
library(CAMERA)
#Single sample
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
xsa <- xsAnnotate(xs)
xsa.grp <- groupDen(xsa, bw=0.5)

#Multiple sample
library(faahKO)
xs <- group(faahko)

#With specific selected sample
xsa <- xsAnnotate(xs, sample=1)
xsa.grp <- groupDen(xsa)

#With automatic selection
xsa.auto <- xsAnnotate(xs)
xsa.grp.auto <- groupDen(xsa.auto)
```

---

groupFWHM

*FWHM-Grouping of LC/ESI-MS data*

---

**Description**

Group peaks of a `xsAnnotate` object according to their retention time into pseudospectra-groups. Uses the peak FWHMs as grouping borders. Returns `xsAnnotate` object with pseudospectra information.

**Usage**

```
groupFWHM(object, sigma = 6 , perfwHM = 0.6, intval = "maxo")
```

**Arguments**

<code>object</code>	the <code>xsAnnotate</code> object
<code>sigma</code>	the multiplier of the standard deviation
<code>perfwHM</code>	percentage of the width of the FWHM
<code>intval</code>	intensity values for ordering. Allowed values are <code>into</code> , <code>maxo</code> , <code>intb</code>



## Details

Every peak who eluate at the same time-point as a selected peak, will be part of the group. Same time-point is defined about the  $Rt_{med} \pm FWHM * perfwhm$ . For a single sample `xcmsSet` the selection of peaks starts at the most abundant and goes down to the smaller ones. With a multiple sample set the automatic selection uses the most abundant peak as an representative for every feature group, according to the `xcms` grouping. With the `xsAnnotate` sample parameter a sample selection can be defined to use only specific samples. See [xsAnnotate-class](#) for further information. The FWHM (full width at half maximum) of a peak is estimated as  $FWHM = SD * 2.35$ . For the calculation of the SD, the peak is assumed as normal distributed.

## Author(s)

Carsten Kuhl <ckuhl@ipb-halle.de>

## Examples

```
library(CAMERA)
#Single sample
file <- system.file('mzML/MM14.mzML', package = "CAMERA")
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5,10))
an <- xsAnnotate(xs)
an <- groupFWHM(an)

#Multiple sample
library(faahKO)
xs <- group(faahko)

#With specific selected sample
xs.anno <- xsAnnotate(xs, sample=1)
xs.group <- groupFWHM(xs.anno)

#With automatic selection
xs.anno.auto <- xsAnnotate(xs)
xs.group.auto <- groupFWHM(xs.anno.auto)
```

---

massWindowSizes

*The supported mass window sizes*

---

## Description

Returns the set of supported mass window sizes for the given compound database

## Usage

```
massWindowSizes(libraryName = "kegg")
```

## Arguments

libraryName    The compound database

**Value**

Vector of supported mass window sizes

**Author(s)**

Hendrik Treutler

**Examples**

```
massWindowSizes()
```

---

mm14

*Extract of marker mixture 14 LC/MS data*

---

**Description**

xcmsSet object containing quantitated LC/MS peaks from a marker mixture. The data is a centroided subset from 117-650 m/z and 271-302 seconds with 134 peaks. Positive ionization mode data in mzML file format.

**Usage**

```
data(mm14)
```

**Format**

The format is:

```
Formal class 'xcmsSet' [package "xcms"] with 8 slots
  @ peaks      : num [1:83, 1:11] 117 117 118 119 136
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
  .. .. ..$ : chr [1:11] "mz" "mzmin" "mzmax" "rt"
  ..@ groups   : logi[0 , 0 ]
  ..@ groupidx : list()
  ..@ phenoData:'data.frame': 1 obs. of 1 variable:
  .. ..$ class: Factor w/ 1 level "mzML": 1
  ..@ rt       :List of 2
  .. ..$ raw    :List of 1
  .. .. ..$ : num [1:112] 270 271 271 271 272 ...
  .. ..$ corrected:List of 1
  .. .. ..$ : num [1:112] 270 271 271 271 272 ...
  ..@ filepaths: chr "mzML/MM14.mzML"
  ..@ profinfo :List of 2
  .. ..$ method: chr "bin"
  .. ..$ step   : num 0.1
  ..@ polarity  : chr(0)
```

**Details**

The corresponding raw mzData files are located in the mzML subdirectory of this package.

**Author(s)**

Carsten Kuhl <ckuhl@ipb-halle.de>

**Source**

<http://doi:10.1186/1471-2105-9-504>

**References**

Data originally reported in "Highly sensitive feature detection for high resolution LC/MS" BMC Bioinformatics; 2008; 9:504.

---

plotEICs-methods

*Plot extracted ion chromatograms from (multiple) Pseudospectra*

---

**Description**

Batch plot a list of extracted ion chromatograms to the current graphics device.

**Arguments**

object	the xsAnnotate object
xraw	xcmsRaw object underlying the the xsAnnotate
maxlabel	How many m/z labels to print
sleep	seconds to pause between plotting EICs
...	other graphical parameters

**Value**

None.

**Methods**

```
object = "xsAnnotate" plotEICs(object, xraw, pspec=1:length(object@pspectra), maxlabel=0,
sleep=0)
```

**Author(s)**

Steffen Neumann, <sneumann@ipb-halle.de>

**See Also**

[xsAnnotate-class](#), [png](#), [pdf](#), [postscript](#),

---

plotPsSpectrum-methods

*Plot a Pseudospectrum*

---

### Description

Plot a pseudospectrum, with the most intense peaks labelled, to the current graphics device.

### Usage

```
plotPsSpectrum(object, pspec=1:length(object@pspectra), log=FALSE, value="into", maxlabel=0, title=)
```

### Arguments

object	the xsAnnotate object
pspec	ID of the pseudospectrum to print
log	Boolean, whether the log(intensity) should be shown
value	Which of a peak's intensities should be used
maxlabel	How many m/z labels to print
title	Main title of the Plot
mzrange	Which m/z range should plotted
sleep	Time (in seconds) to wait between successive Spectra, if multiple pspec are requested.
cexMulti	Cex multiplier for peak labels
...	Additional parameter for function plot

### Value

None.

### Methods

```
signature(object = "xsAnnotate") object deriviving from class "xsAnnotate"
```

### Author(s)

Steffen Neumann, <sneumann@ipb-halle.de>

### See Also

[xsAnnotate-class](#), [png](#), [pdf](#), [postscript](#),

---

psDist-methods                      *Distance methods for xsAnnotate*

---

## Description

The package `xcms` contains several methods for calculating a distance between two sets of peaks. the CAMERA method `psDist` is the generic wrapper to use these methods for processing two pseudospectra from two different `xsAnnotate` objects.

## Arguments

<code>object1</code>	a <code>xsAnnotate</code> object with pseudospectra
<code>object2</code>	a <code>xsAnnotate</code> object with pseudospectra
<code>PSpec1</code>	index of pseudospectrum in <code>object1</code>
<code>PSpec2</code>	index of pseudospectrum in <code>object2</code>
<code>method</code>	method to use for distance calculation. See details.
<code>...</code>	<code>mzabs</code> , <code>mzppm</code> and parameters for the distance function.

## Details

Different algorithms can be used by specifying them with the `method` argument. For example to use the "meanMZmatch" approach one would use: `specDist(object1, object2, pspectrum1, pspectrum2, method="meanMZmatch")`. This is also the default.

Further arguments given by `...` are passed through to the function implementing the method.

A character vector of *nicknames* for all the algorithms which are available is returned by `getOption("BioC")$xcms$specDist`. If the nickname of a method is called "meanMZmatch", the help page for that specific method can be accessed with `?specDist.meanMZmatch`.

## Value

<code>mzabs</code>	maximum absolute deviation for two matching peaks
<code>mzppm</code>	relative deviations in ppm for two matching peaks
<code>symmetric</code>	use symmetric pairwise m/z-matches only, or each match

## Methods

**object1 = "xsAnnotate"** `specDist(object1, object2, pspectrum1, pspectrum2, method,...)`

## Author(s)

Joachim Kutzera, <jkutzer@ipb-halle.de>

---

`pspec2metfrag`*Export the putative fragments as MetFrag query files*

---

## Description

MetFrag is an in-silico metabolite identification system, which aims to putatively identify compounds from fragmentation MS data, especially from tandem-MS, but also in-source fragments might give additional hints on top of the accurate mass of the precursor alone.

## Usage

```
pspec2metfrag(object, pspecidx=NULL, filedir=NULL)
pspec2metfusion(object, pspecidx=NULL, filedir=NULL)
```

## Arguments

<code>object</code>	an <code>xsAnnotate</code> object
<code>pspecidx</code>	Index of pspectra to export, if <code>NULL</code> then all are exported.
<code>filedir</code>	Directory for placement of batch query files

## Details

For each spectrum in `pspecidx` (or all in the `xsAnnotate` object), for each [M] mass hypothesis, remove all non-fragment peaks (isotopes, clusters, adducts) and pass them to MetFrag and MetFusion batch query files.

## Value

Returns a list

## Author(s)

Carsten Kuhl <ckuhl@ipb-halle.de>

## Examples

```
library(CAMERA)
file <- system.file('mzML/MM14.mzML', package = "CAMERA");
xs <- xcmsSet(file, method="centWave", ppm=30, peakwidth=c(5, 10));
an <- xsAnnotate(xs);
an <- groupFWHM(an);
an <- findIsotopes(an); #optional step
an <- findAdducts(an, polarity="positive")

pspec2metfrag(an, pspecidx=c(1))
```

---

ruleSet	Class ruleSet
---------	---------------

---

### Description

The class ruleSet is used to read lists of ions, adducts and neutral losses, and compile the dynamic ruleSet from those. This makes it possible to modify the default rules for certain analytical settings.

### Slots

**ionlistfile:** File of known charged ions, an example is found in CAMERA/lists/ions.csv .  
**neutrallossfile:** File of known neutral losses, an example is found in CAMERA/lists/neutralloss.csv.  
**neutraladditionfile:** File of known adducts, an example is found in CAMERA/lists/lists/neutraladdition.csv .  
**ionlist:** Known charged ions.  
**neutralloss:** Known neutral losses.  
**neutraladdition:** Known adducts.  
**maxcharge:** .  
**mol:** .  
**nion:** .  
**nnloss:** .  
**nnadd:** .  
**nh:** .  
**polarity:** Polarity of the ruleSet.  
**rules:** data.frame of resulting mass differences, this is the dynamic ruleSet.  
**lib.loc** Path to local R library

### Extends

Class "[Versioned](#)", directly.

### Methods

Methods implemented for ruleSet

**setDefaultLists** signature(object = "ruleSet"): Set filenames for the lists shipped with CAMERA.

**readLists** signature(object = "ruleSet"): Read and parse the lists from the files.

**setDefaultParams** signature(object = "ruleSet"): Set the default parameters for rule generation.

**setParams** signature(object = "ruleSet"): Set the parameters for rule generation.

**generateRules** signature(object = "ruleSet"): Create the rules in ruleSet@rules .

**Author(s)**

Steffen Neumann and Carsten Kuhl

**Examples**

```
r <- new("ruleSet");
r2 <- setDefaultLists(r) ;
r3 <- readLists(r2) ;
r4 <- setDefaultParams(r3) ;
r5 <- generateRules(r4)
dim(r5@rules)
```

---

`xsAnnotate`*xsAnnotate constructor for an provided xcmsSet object*

---

**Description**

This function deals with the construction of an `xsAnnotate` object. It extracts the peaktable from a provided `xcmsSet`, which is used for all further analysis. The `xcmsSet` can be a single sample or multiple sample experiment. Since some functions needs the raw data a selection algorithm must be chosen in the case of a multiple sample. CAMERA includes two different strategies: A defined selection of samples (`sample = indices of samples`) or the default automatic solution (`sample = NA`). The automatic solution chooses the best sample for a specific groups called pseudospectrum, see [groupFWHM](#) and [groupCorr](#). It returns a `xsAnnotate` object, see [xsAnnotate-class](#).

**Usage**

```
xsAnnotate(xs = NULL, sample=NA, nSlaves = 1, polarity = NULL)
```

**Arguments**

<code>xs</code>	a <code>xcmsSet</code> object
<code>sample</code>	Indices of the group <code>xcmsSet</code> sample, that are used for the EIC correlation step. For automatic selection don't set a value. For use all samples simply define <code>sample = c(1:n)</code> , with <code>n = number of samples</code> .
<code>nSlaves</code>	For parallel mode set <code>nSlaves</code> higher than 1, but not higher than the number of cpu cores.
<code>polarity</code>	Set polarity mode: "positive" or "negative"

**Value**A `xsAnnotate` object.**Author(s)**

Carsten Kuhl, &lt;ckuhl@ipb-halle.de&gt;



**See Also**[xsAnnotate-class](#)**Examples**

```
library(faahKO)
xs <- group(faahko)
xsa <- xsAnnotate(xs, sample=c(1:12))

#With automatic selection
xsa.autoselect <- xsAnnotate(xs)
```

---

xsAnnotate-class	<i>Class xsAnnotate, a class for annotated peak data</i>
------------------	----------------------------------------------------------

---

**Description**

This class transforms a [xcmsSet](#) object with peaks from multiple LC/MS or GC/MS samples into a set of annotation results. It contains searching algorithms for isotopes and adducts, peak grouping algorithms to find connected peak, which originate from the same molecule.

**Objects from the Class**

Objects can be created with the [xsAnnotate](#) constructor which include the peaktable from a provided [xcmsSet](#). Objects can also be created by calls of the form `new("xsAnnotate", ...)`.

**Slots**

**annoGrp:** Assignment of mass hypotheses to correlation groups  
**annoID:** The assignemnt of peaks to the mass difference rule used  
**derivativeIons:** List with annotation result for every peak  
**formula:** Matrix containing putative sum formula (intended for future use)  
**isoID:** Matrix containing IDs and additional of all annotated isotope peaks  
**groupInfo:** (grouped) Peaktable with "into" values  
**isotopes:** List with annotated isotopid results for every peak  
**polarity:** A single string with the polarity mode of the peaks  
**pspectra:** List contains all pseudospectra with there peak IDs  
**psSamples:** List containing information with sample was sample was selecteted as representative (automatic selection)  
**ruleset:** A dataframe describing the mass difference rules used for the annotation  
**runParallel:** Flag if CAMERA runs in serial or parallel mode  
**sample:** Number of the used xcmsSet sample (beforehand sample selection)  
**xcmsSet:** The embedded xcmsSet

**Methods**

**groupFWHM** signature(object = "xsAnnotate"): group the peak data after the FWHM of the retention time

**groupCorr** signature(object = "xsAnnotate"): group the peak data after the correlation of the EICs

**findIsotopes** signature(object = "xsAnnotate"): search for possible isotopes in the spectra

**findAdducts** signature(object = "xsAnnotate"): search for possible adducts in the spectra

**plotEICs** signature(object = "xsAnnotate"): plot EICs of pseudospectra

**Note**

No notes yet.

**Author(s)**

Carsten Kuhl, <ckuhl@ipb-halle.de>

**See Also**

[xsAnnotate](#)

# Index

- \* **classes**
  - compoundQuantiles-class, 15
  - ruleSet, 39
  - xsAnnotate-class, 41
- \* **datasets**
  - mm14, 34
- \* **file**
  - xsAnnotate, 40
- \* **hplot**
  - plotEICs-methods, 35
  - plotPsSpectrum-methods, 36
- \* **methods**
  - annotate-methods, 3
  - annotateDiffreport, 4
  - calcCaS-methods, 6
  - calcCiS-methods, 7
  - calcIsotopes-methods, 8
  - calcPC-methods, 9
  - calcPC.hcs, 10
  - calcPC.lpc, 11
  - cleanParallel, 12
  - combinexsAnnos, 12
  - findAdducts-methods, 16
  - findIsotopes, 17
  - findIsotopesWithValidation, 18
  - findKendrickMasses, 19
  - findNeutralLoss, 20
  - findNeutralLossSpecs, 21
  - getAllPeakEICs, 22
  - getIsotopeCluster, 24
  - getPeaklist, 26
  - getpspectra, 27
  - getReducedPeaklist, 28
  - groupCorr, 29
  - groupDen, 31
  - groupFWHM, 32
  - plotEICs-methods, 35
  - plotPsSpectrum-methods, 36
  - psDist-methods, 37
  - pspec2metfrag, 38
  - annotate (annotate-methods), 3
  - annotate, xcmsSet-method (annotate-methods), 3
  - annotate-methods, 3
  - annotateDiffreport, 4
  - annotateDiffreport, xsAnnotate-methods (annotateDiffreport), 4
  - calcCaS, 8, 30
  - calcCaS (calcCaS-methods), 6
  - calcCaS, xsAnnotate-method (calcCaS-methods), 6
  - calcCaS-methods, 6
  - calcCiS, 7, 30
  - calcCiS (calcCiS-methods), 7
  - calcCiS, xsAnnotate-method (calcCiS-methods), 7
  - calcCiS-methods, 7
  - calcIsotopes (calcIsotopes-methods), 8
  - calcIsotopes, xsAnnotate-method (calcIsotopes-methods), 8
  - calcIsotopes-methods, 8
  - calcPC, 7-9, 11, 30
  - calcPC (calcPC-methods), 9
  - calcPC, calcPC-method (calcPC-methods), 9
  - calcPC, xsAnnotate-method (calcPC-methods), 9
  - calcPC-methods, 9
  - calcPC.hcs, 10, 10
  - calcPC.hcs, xsAnnotate-method (calcPC.hcs), 10
  - calcPC.lpc, 10, 11
  - calcPC.lpc, xsAnnotate-method (calcPC.lpc), 11
  - class:ruleSet (ruleSet), 39
  - cleanParallel, 12
  - combinexsAnnos, 12
  - compoundLibraries, 13

- compoundQuantiles, [14](#), [15](#), [16](#)
- compoundQuantiles-class, [15](#)
- diffreport, [5](#)
- findAdducts, [42](#)
- findAdducts (findAdducts-methods), [16](#)
- findAdducts, xsAnnotate-method (findAdducts-methods), [16](#)
- findAdducts-methods, [16](#)
- findIsotopes, [8](#), [17](#), [19](#), [42](#)
- findIsotopes, xsAnnotate-method (findIsotopes), [17](#)
- findIsotopesWithValidation, [18](#)
- findIsotopesWithValidation, xsAnnotate-method (findIsotopesWithValidation), [18](#)
- findKendrickMasses, [19](#)
- findNeutralLoss, [20](#)
- findNeutralLossSpecs, [21](#)
- generateRules (ruleSet), [39](#)
- generateRules, ruleSet-method (ruleSet), [39](#)
- getAllPeakEICs, [7](#), [8](#), [22](#)
- getAllPeakEICs, xsAnnotate-method (getAllPeakEICs), [22](#)
- getAtomCount, [15](#), [16](#)
- getAtomCount (getAtomCount, compoundQuantiles-method), [23](#)
- getAtomCount, compoundQuantiles-method, [23](#)
- getIsotopeCluster, [24](#)
- getIsotopeProportion, [16](#)
- getIsotopeProportion (getIsotopeProportion, compoundQuantiles-method), [25](#)
- getIsotopeProportion, compoundQuantiles-method, [15](#), [25](#)
- getPeaklist, [24](#), [26](#), [29](#), [30](#)
- getPeaklist, xsAnnotate-method (getPeaklist), [26](#)
- getpspectra, [27](#)
- getReducedPeaklist, [28](#)
- getReducedPeaklist, xsAnnotate-method (getReducedPeaklist), [28](#)
- groupCorr, [7](#), [8](#), [10](#), [11](#), [29](#), [40](#), [42](#)
- groupCorr, xsAnnotate-method (groupCorr), [29](#)
- groupDen, [31](#)
- groupDen, xsAnnotate-method (groupDen), [31](#)
- groupFWHM, [9–11](#), [32](#), [40](#), [42](#)
- groupFWHM, xsAnnotate-method (groupFWHM), [32](#)
- highlyConnSG, [11](#)
- label.propagation.community, [11](#)
- massWindowSizes, [33](#)
- mm14, [34](#)
- pdf, [35](#), [36](#)
- plotEICs, [42](#)
- plotEICs (plotEICs-methods), [35](#)
- plotEICs, xsAnnotate-method (plotEICs-methods), [35](#)
- plotEICs-methods, [35](#)
- plotEICs.xsAnnotate (plotEICs-methods), [35](#)
- plotPsSpectrum (plotPsSpectrum-methods), [36](#)
- plotPsSpectrum, xsAnnotate-method (plotPsSpectrum-methods), [36](#)
- plotPsSpectrum-methods, [36](#)
- plotPsSpectrum.xsAnnotate (plotPsSpectrum-methods), [36](#)
- png, [35](#), [36](#)
- postscript, [35](#), [36](#)
- psDist (psDist-methods), [37](#)
- psDist, CAMERA-method (psDist-methods), [37](#)
- psDist-methods, [37](#)
- pspec2metfrag, [38](#)
- pspec2metfusion (pspec2metfrag), [38](#)
- readLists (ruleSet), [39](#)
- readLists, ruleSet-method (ruleSet), [39](#)
- ruleSet, [39](#)
- ruleSet-class (ruleSet), [39](#)
- setDefaultLists (ruleSet), [39](#)
- setDefaultLists, ruleSet-method (ruleSet), [39](#)
- setDefaultParams (ruleSet), [39](#)

setDefaultParams, ruleSet-method  
    (ruleSet), [39](#)

setParams (ruleSet), [39](#)

setParams, ruleSet, numeric, numeric, numeric, numeric, numeric, numeric, character, character-method  
    (ruleSet), [39](#)

show, ruleSet-method (ruleSet), [39](#)

show, xsAnnotate-method  
    (xsAnnotate-class), [41](#)

Versioned, [39](#)

xcmsSet, [41](#)

xsAnnotate, [40](#), [41](#), [42](#)

xsAnnotate-class, [3](#), [4](#), [40](#), [41](#)