

Package ‘LipidMS’

November 3, 2025

Type Package

Title Lipid Annotation for LC-MS/MS DDA or DIA Data

Version 3.1.1

Maintainer M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Description Lipid annotation in untargeted LC-MS lipidomics based on fragmentation rules. Alcoriza-Balaguer MI, Garcia-Canaveras JC, Lopez A, Conde I, Juan O, Carretero J, Lahoz A (2019) <[doi:10.1021/acs.analchem.8b03409](https://doi.org/10.1021/acs.analchem.8b03409)>.

Depends R (>= 4.0), ggplot2, future, future.apply

License GPL (>= 2)

LazyData TRUE

RoxygenNote 7.3.3

Imports readMzXmlData, utils, CHNOSZ, scales, stats, graphics, grDevices, iterators, patchwork, DT, R.utils, parallel, parallelly

Encoding UTF-8

Suggests knitr, rmarkdown, shiny, shinythemes, shinyjs, plotly

VignetteBuilder knitr

NeedsCompilation yes

Author M Isabel Alcoriza-Balaguer [aut, cre],
Celia Regal-Marques [aut]

Repository CRAN

Date/Publication 2025-11-03 13:00:03 UTC

Contents

acylcerdb	4
adductsTable	5
alignmsbatch	5
annotatemsbatch	7
assignDB	8

baconjdb	9
badb	9
batchdataProcessing	10
carnitinesdb	12
CEdb	13
cerdb	13
cerPdb	14
chainFrgs	14
checkClass	15
checkIntensityRules	17
cldb	18
coelutingFrgs	18
coelutionScore	19
combineChains	19
confLevels	20
createLipidDB	21
crossTables	22
dataProcessing	22
ddaFrgs	24
dgdb	25
fadb	26
fahfadb	26
fillpeakmsbatch	27
findCandidates	28
getInclusionList	29
groupmsbatch	30
hfadb	31
idAcylCerneg	32
idAcylCerpos	34
idBAneg	37
idCarpos	39
idCEpos	41
idCerneg	43
idCerPneg	45
idCerpos	47
idCerPpos	50
idCLneg	52
idDGpos	54
idFAHFAneg	57
idFAneg	59
idLPCneg	61
idLPCpos	63
idLPEneg	65
idLPEpos	67
idLPGneg	69
idLPIneg	71
idLPSneg	73
idMGpos	75

idNEG	77
idPCneg	78
idPConeg	80
idPCopos	83
idPCpneg	85
idPCpos	88
idPCppos	90
idPEneg	92
idPEoneg	95
idPEopos	97
idPEpneg	99
idPEpos	102
idPEppos	104
idPGneg	106
idPGpos	109
idPIneg	111
idPIpos	113
idPOS	116
idPSneg	117
idSMneg	119
idSMpos	122
idSphneg	124
idSphPneg	126
idSphpos	128
idSphPpos	130
idTGpos	132
LipidMSapp	134
lysopadb	135
lysopaodb	135
lysopcdb	136
lysopcodb	136
lysopcpdb	137
lysopedb	137
lysopeodb	138
lysopepdb	138
lysopgdb	139
lysopidb	139
lysopsdb	140
mgdb	140
nlsphdb	141
organizeResults	141
padb	142
pcdb	143
pcodb	143
pcpdb	144
pedb	144
peodb	145
pepdb	145

pgdb	146
pidb	146
ploteicmsbatch	147
plotLipids	147
plotticmsbatch	148
psdb	149
rtdevplot	149
searchIsotopes	150
searchIsotopesmsbatch	151
setmsbatch	152
smdb	153
sphdb	153
sphPdb	154
tgdb	154
Index	155

acylcerdb	<i>AcylCeramides database</i>
-----------	-------------------------------

Description

In silico generated database for common acylceramides.

Usage

```
data("acylcerdb")
```

Format

Data frame with 192 observations and the following 3 variables.

- formula character vector containing molecular formulas.
- total character vector indicating the total number of carbons and double bounds of the chains.
- Mass numeric vector with the neutral masses.

adductsTable	<i>Adducts table</i>
--------------	----------------------

Description

Table of possible adducts to be employed by LipidMS and related information.

Usage

```
data("adductsTable")
```

Format

Data frame with 18 observations and the following 4 variables.

adduct character vector with the adducts names.

mdiff numeric vector indicating the mass differences.

charge numeric vector indicating the charge.

n numeric vector. It indicates if the ion is a monomer (1), a dimer (2), etc.

alignmsbatch	<i>Align samples from an msbatch</i>
--------------	--------------------------------------

Description

Align samples from an msbatch to correct time drifts during acquisition queues.

Usage

```
alignmsbatch(  
  msbatch,  
  dmz = 5,  
  drt = 30,  
  minsamples,  
  minsamplesfrac = 0.75,  
  span = 0.4,  
  parallel = FALSE,  
  ncores,  
  global_gb = getOption("LipidMS.future.globals.maxSizeGB", 24),  
  verbose = TRUE  
)
```

Arguments

msbatch	msbatch obtained from the setmsbatch function.
dmz	mass tolerance between peak groups in ppm.
drt	maximum rt distance between peaks for alignment in seconds.
minsamples	minimum number of samples represented in each cluster used for the alignment.
minsamplesfrac	minimum samples fraction represented in each cluster used for the alignment. Used to calculate minsamples in case it is missing.
span	span parameter for loess rt deviation smoothing.
parallel	logical. If TRUE, parallel processing will be performed.
ncores	number of cores to be used in case parallel is TRUE.
global_gb	numeric. Gigabytes to set as future.globals.maxSize <i>**inside**</i> the function. Defaults to 'getOption("LipidMS.future.globals.maxSizeGB", 24)'.
verbose	print information messages.

Details

First, peak partitions are created based on the *enviPick* algorithm to speed up the following clustering algorithm. Briefly, peaks are ordered increasingly by *mz* and *RT* and grouped based on user-defined tolerances (*dmz* and *drt*). Each peak is initialized as a partition and then, they are evaluated to decide whether or not they can be joined to the previous partition. If *mz* and *RT* of a peak matches tolerance of any of the peaks in the previous partition, it is reassigned. Then, clustering algorithm is executed to group peaks based on their *RT* following the next steps for each partition:

1. Each peak in the partition is initialized as a new cluster. For each cluster we will keep the minimum, maximum and mean value of the *RT*, which at this point have the same values.
2. Calculate a distance matrix between all clusters. This distance will be the greatest difference between minimum and maximum values of each cluster. Distances between clusters which share peaks from the same samples will be set to NA.
3. While any distance is different to NA, search the minimum distance between two clusters.
4. If distance is below the maximum distance allowed, join clusters and update minimum, maximum and mean values, else, set distance to NA and go back to point 3.

Then, clusters with a sample representation over *minsamples* or *minsamplesfrac*, will be used for alignment. To this end, an *RT* matrix is built containing the *RT* of the peaks for each sample from the selected clusters. Then, median *RT* is calculated for each cluster and an *RT* deviation matrix is obtained. Finally, time drifts for each sample are corrected using loess regression by constructing a function based on *RT* deviation and median.

Value

aligned msbatch

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafe.es>

References

Partitioning algorithm has been imported from *enviPick* R-package: [https://cran.r-project.org/web/packages/enviPick/index.h](https://cran.r-project.org/web/packages/enviPick/index.html)

Examples

```
## Not run:
msbatch <- alignmsbatch(msbatch)

## End(Not run)
```

annotatemsbatch	<i>Lipid annotation for an msbatch</i>
-----------------	--

Description

Summarize annotation results of an msbatch into the feature table

Usage

```
annotatemsbatch(
  msbatch,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 5,
  coelCutoff = 0.8,
  lipidClassesPos = c("MG", "LPC", "LPE", "PC", "PCo", "PCp", "PE", "PEo", "PEp", "PG",
    "PI", "Sph", "SphP", "Cer", "CerP", "AcylCer", "SM", "Carnitine", "CE", "DG", "TG"),
  lipidClassesNeg = c("FA", "FAHFA", "LPC", "LPE", "LPG", "LPI", "LPS", "PC", "PCo",
    "PCp", "PE", "PEo", "PEp", "PG", "PI", "PS", "Sph", "SphP", "Cer", "CerP", "AcylCer",
    "SM", "CL", "BA"),
  dbs,
  simplifyAnnotations = FALSE,
  parallel = FALSE,
  ncores,
  global_gb = getOption("LipidMS.future.globals.maxSizeGB", 24)
)
```

Arguments

msbatch	msbatch
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 5 seconds.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
lipidClassesPos	classes of interest in ESI+.

lipidClassesNeg
 classes of interest in ESI-.

dbs
 list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See [createLipidDB](#) and [assignDB](#).

simplifyAnnotations
 logical. If TRUE, only the most frequent id will be kept (recommended when only pool samples have been acquired in DIA or DDA). If FALSE, all annotations will be shown.

parallel
 logical.

ncores
 number of cores to be used in case parallel is TRUE.

global_gb
 numeric. Gigabytes to set as future.globals.maxSize ****inside**** the function. Defaults to 'getOption("LipidMS.future.globals.maxSizeGB", 24)'.

Value

msbatch

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Examples

```
## Not run:
msbatch <- annotatemsbatch(msbatch)

msbatch$features

## End(Not run)
```

assignDB

Load LipidMS default data bases

Description

load all LipidMS default data bases required to run identification functions.

Usage

```
assignDB()
```

Value

list of data frames

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Examples

```
## Not run:  
dbs <- assignDB()  
  
## End(Not run)
```

baconjdb	<i>Bile acids conjugates database</i>
----------	---------------------------------------

Description

Common bile acids conjugates. It can be modified to look for other BA species.

Usage

```
data("baconjdb")
```

Format

Data frame with 2 observations and the following 2 variables.

total character vector indicating the names of the conjugates.

Mass numeric vector with the neutral masses of the conjugates fragments.

badb	<i>Bile acids database</i>
------	----------------------------

Description

In silico generated database for common bile acids.

Usage

```
data("badb")
```

Format

Data frame with 9 observations and the following 5 variables.

formula character vector with the molecular formulas.

total character vector containing the names of the BAs (i.e. CA, TDCA, GLCA...).

Mass numeric vector with the neutral masses.

conjugate character vector containing the conjugate of each BA.

base character vector containing the core of each BA.

batchdataProcessing	<i>Process several mzXML files (peakpicking and isotope annotation) and create an msbatch for batch processing.</i>
---------------------	---

Description

Process several mzXML files (peakpicking and isotope annotation) and create an msbatch for batch processing.

Usage

```
batchdataProcessing(
  files,
  metadata,
  polarity,
  dmzagglom = 15,
  drtagglom = 500,
  drtclust = 100,
  minpeak = c(5, 3),
  drtgap = 10,
  drtminpeak = 15,
  drtmaxpeak = c(100, 200),
  recurs = 5,
  sb = c(3, 2),
  sn = 2,
  minint = c(1000, 100),
  weight = c(2, 3),
  dmzIso = 10,
  drtIso = 5,
  parallel = FALSE,
  ncores,
  global_gb = getOption("LipidMS.future.globals.maxSizeGB", 24),
  verbose = TRUE
)
```

Arguments

files	file paths of the mzXML files. Optional.
metadata	csv file or data.frame with 3 columns: sample (samples named as the mzXML files), acquisitionmode (MS, DIA or DDA) and groups (i.e. blank, QC, sample). DIA, DDA and MS files are allowed, but only DIA and DDA files will be used for lipid annotation.
polarity	character value: negative or positive.
dmzagglom	mz tolerance (in ppm) used for partitioning and clustering.
drtagglom	rt window used for partitioning (in seconds).
drtclust	rt window used for clustering (in seconds).
minpeak	minimum number of measurements required for a peak.
drtgap	maximum RT gap length to be filled (in seconds).
drtminpeak	minimum RT width of a peak (in seconds). At least minpeak within the drtmin-peak window are required to define a peak.
drtmaxpeak	maximum RT width of a single peak (in seconds).
recurs	maximum number of peaks within one EIC.
sb	signal-to-base ratio.
sn	signal-to-noise ratio.
minint	minimum intensity of a peak.
weight	weight for assigning measurements to a peak.
dmzIso	mass tolerance for isotope matching.
drtIso	time windows for isotope matching.
parallel	logical.
ncores	number of cores to be used in case parallel is TRUE.
global_gb	numeric. Gigabytes to set as future.globals.maxSize **inside** the function. Defaults to 'getOption("LipidMS.future.globals.maxSizeGB", 24)'.
verbose	print information messages.

Details

This function executes 2 steps: 1) creates an msubject for each sample (using the [dataProcessing](#) function) and 2) sets an msbatch ([setmsbatch](#) function).

Numeric arguments accept one or two values for MS1 and MS2, respectively.

Value

msbatch

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafe.es>

References

Peak-picking algorithm has been imported from enviPick R-package: <https://cran.r-project.org/web/packages/enviPick/index>.

See Also

[dataProcessing](#) and [setmsbatch](#)

Examples

```
## Not run:
# if metadata is a data frame:
msbatch <- batchdataProcessing(metadata$sample, metadata, polarity = "positive",
dmzagglom = 25, drtagglom = 500, drtclust = 60, minpeak = c(5, 3),
drtgap = 5, drtminpeak = 20, drtmaxpeak = 100, recurs = 5, sb = c(3, 2),
sn = 2, minint = c(1000, 100), weight = 2, dmzIso = 10, drtIso = 5)

# if metadata is a csv file:
msbatch <- batchdataProcessing(metadata = "metadata.csv", polarity = "positive",
dmzagglom = 25, drtagglom = 500, drtclust = 60, minpeak = c(5, 3),
drtgap = 5, drtminpeak = 20, drtmaxpeak = 100, recurs = 5, sb = c(3, 2),
sn = 2, minint = c(1000, 100), weight = 2, dmzIso = 10, drtIso = 5)

## End(Not run)
```

carnitinesdb

Carnitines database

Description

In silico generated database for common carnitines.

Usage

```
data("carnitinesdb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

CEdb*CEs database*

Description

In silico generated database for common CEs.

Usage

```
data("CEdb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

cerdb*Ceramides database*

Description

In silico generated database for common ceramides.

Usage

```
data("cerdb")
```

Format

Data frame with 52 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

cerPdb	<i>Ceramides Phosphate database</i>
--------	-------------------------------------

Description

In silico generated database for common ceramides phosphate.

Usage

```
data("cerPdb")
```

Format

Data frame with 52 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

chainFrgs	<i>Search of chain specific fragments</i>
-----------	---

Description

Search of specific fragments that inform about the chains structure.

Usage

```
chainFrgs(coelfrgs, chainfrags, ppm = 10, candidates, f = NULL, dbs)
```

Arguments

coelfrgs	coeluting fragments for each candidate. Output of coelutingFrgs .
chainfrags	character vector containing the fragmentation rules for the chain fragments. If it is an empty vector, chains will be calculated based on the difference between the precursor and the other chain. See details.
ppm	m/z tolerance in ppm.
candidates	candidates data frame. If any chain needs to be calculated based on the difference between the precursor and the other chain, this argument will be required. Output of chainFrgs .
f	known chains. If any chain needs to be calculated based on the difference between the precursor and the other chain, this argument will be required. Output of chainFrgs .
dbs	list of data bases required for the annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be changed. If data bases have been customized using createLipidDB , they also have to be modified here.

Details

The chainfrags argument must contain the fragmentation rules which inform about the chains structure. For example, in the case of PG subclass, the chain in sn1 position is identified by the lysoPG as M-H resulting from the loss of the FA chain of sn2; and the chain in sn2 position is identified as the free FA chain as M-H. These two fragments need to be searched in two different steps: in the first step we will look for lysoPGs coeluting with the precursor using chainfrags = c("lysopg_M-H"); then, we will look for FA chains using chainfrags = c("fa_M-H"). This information can be combined later using [combineChains](#) function.

To indicate the fragments to be searched, the class of lipid is written using the same names as the LipidMS databases without the "db" at the end (i.e. pa, dg, lysopa, mg, CE, etc.), and the adduct has to be indicated as it appears in the adductsTable, both parts separated by "_". In case some chain needs to be searched based on a neutral loss, this can be defined using "NL-" prefix, followed by the database and adduct. If this neutral loss is employed to find the remaining chain, "cbdiff-" prefix allows to calculate the difference in carbons and double bounds between the precursor and the building block found. For example, "cbdiff-dg_M+H-H2O" will look for DG as M+H-H2O and then, it will return the difference between their number of carbons and double bounds and the ones from the precursor. Otherwise, "NL-mg_M+H-H2O" will look for fragments coming from the loss of MGs.

In case these fragments identified as losses from the precursors are going to be employed for the intensity rules, this same prefix has to be added.

If a chain is calculated based on the difference of total number of carbons and double bounds between the precursor and a previously searched chain, chainfrags argument must be a character vector c("") and candidates data frame and chain fragments list must be provided.

Value

List of data frames with the chain fragments found.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

checkClass

Search of class fragments to confirm the lipid class.

Description

Search of characteristic fragments that confirm a given lipid class.

Usage

```
checkClass(candidates, coelfrags, clfrags, ftype, clrequisites, ppm = 10, dbs)
```

Arguments

candidates	output of findCandidates function.
coelfrags	list of peaks coeluting with each candidate. Output of coelutingFrgs .
clfrags	vector containing the expected fragments for a given lipid class. See details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See details.
clrequisites	logical vector indicating if each class fragment is required or not. If none of the fragment is required, at least one of them must be present within the coeluting fragments. If the presence of any fragment excludes the class, it can be specified by using "excluding".
ppm	m/z tolerance in ppm.
db	list of data bases required for the annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be changed. If data bases have been customized using createLipidDB , they also have to be modified here. It is employed when some fragment belongs to "BB" ftype.

Details

clfrags, ftype and clrequisites will indicate the rules to confirm a lipid class. All three arguments must have the same length.

This function allows three different types of fragments: fragments with a specific m/z as for example 227.0326 for PG in negative mode, which needs to be defined as clfrags = c(227.0326) and ftype = c("F"); neutral losses such as the head group of some PL (i.e. NL of 74.0359 in PG in negative mode), which will be defined as clfrags = c(74.0359) and ftype = c("NL"); or building blocks resulting from the loss of some groups, as for example, PA as M-H resulting from the loss of the head group (glycerol) in PG in ESI-, which will be defined as clfrags = c("pa_M-H") and ftype = c("BB"). The last two options could define the same fragments. In this case just one of them would be necessary.

When using the third type of fragment ("BB"), the building block will be specified in lower case (i.e. pa, dg, lysopa, mg, etc.) and the adduct will be given as it appears in the adductsTable, both separated by "_". Names for the building blocks are the ones used for the LipidMS databases without the "db" at the end.

In case the presence of a fragment indicates that the candidate does not belong to the lipid class (i.e. loss of CH₃ in PE, which corresponds to a PC actually), this will be specified by using clrequisites = c("excluding").

Value

List with 2 elements: a matrix with logical values (presence/absence) of each expected fragment (columns) for each candidate (rows), and a logical vector with the confirmation of the lipid class for each candidate.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

checkIntensityRules	<i>Check intensity rules</i>
---------------------	------------------------------

Description

Check intensity rules to confirm chains position.

Usage

```
checkIntensityRules(inrules, rates, intrequired, nchains, combinations)
```

Arguments

inrules	character vector specifying the fragments to compare. See details.
rates	character vector with the expected rates between fragments given as a string (i.e. "3/1"). See details.
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
nchains	number of chains of the targeted lipid class.
combinations	output of combineChains .

Details

This function will be employed when the targeted lipid class has more than one chain.

Taking PG subclass as an example, intensities of lysoPG fragments (informative for sn1) can be employed to confirm the chains structure (inrules = c("lysopg_sn1/lysopg_sn1")). In this case, the intensity of the lysoPG resulting from the loss of the FA chain in sn2 is at least 3 times greater (rates = c("3/1")) than the lysoPG resulting from the loss of the FA chain in sn1.

For the intrules argument, "/" will be use to separate the fragments related to each chain (sn1/sn2/etc), and "_" will be use to indicate the list in which they'll be searched. This will depend on the chain fragments rules defined previously. Following the example, as we use lysoPG to define the sn1 position, both fragments will be searched in this list (sn1).

For classes with more than one FA chain, if some intensity rule should be employed to identify their position but they are no defined yet, use "Unknown". If it is not necessary because the fragmentation rules are informative enough to define the position (i.e. sphingolipid species), just leave an empty vector.

Value

List of logical vectors with the confirmation for each combination.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

cldb	<i>Cardiolipins database</i>
------	------------------------------

Description

In silico generated database for commo CLs.

Usage

```
data("cldb")
```

Format

Data frame with 714 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

coelutingFrgs	<i>Coeluting fragments extraction</i>
---------------	---------------------------------------

Description

Given a RT and a list of peaks, this function subsets all coeluting fragments within a rt windows. It is used by identification functions to extract coeluting fragments from high energy functions for candidate precursor ions.

Usage

```
coelutingFrgs(
  precursors,
  products,
  rttol,
  rawData = data.frame(),
  coelCutoff = 0
)
```

Arguments

<code>precursors</code>	candidates data frame. Output of findCandidates .
<code>products</code>	peaklist for MS2 function (MSMS).
<code>rttol</code>	rt window in seconds.
<code>rawData</code>	raw scans data. Output of dataProcessing function (MSMS\$rawData).
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied.

Value

List of data frames with the coeluting fragments for each candidate.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

coelutionScore	<i>calculate coelution score between two peaks</i>
----------------	--

Description

Calculate coelution score between two peaks.

Usage

```
coelutionScore(peak1, peak2, rawData)
```

Arguments

peak1	character vector specifying the peakID of the first peak.
peak2	character vector specifying the peakID of the second peak.
rawData	data frame with raw data for each scan. it need to have at least 5 columns: mz, RT, int, Scan (ordinal number for a given MS function) and peakID (peakID to which it has been assigned). #' @keywords internal

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

combineChains	<i>Combine chain fragments that could belong to the same precursor.</i>
---------------	---

Description

It calculates combinations of chain fragments that sum up the same number of carbons and double bounds as the precursor.

Usage

```
combineChains(candidates, nchains, sn1, sn2, sn3, sn4)
```

Arguments

candidates	candidates data frame. Output of findCandidates .
nchains	number of chains of the targeted lipid class.
sn1	list of chain fragments identified for sn1 position. Output of chainFragments .
sn2	list of chain fragments identified for sn2 position. Output of chainFragments . If required.
sn3	list of chain fragments identified for sn3 position. Output of chainFragments . If required.
sn4	list of chain fragments identified for sn4 position. Output of chainFragments . If required.

Value

List of data frames with candidate chains structures.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

confLevels	<i>Confidence Annotation Levels</i>
------------	-------------------------------------

Description

Confidence annotation levels and their hierarchy.

Usage

```
data("confLevels")
```

Format

Data frame with 5 observations and 2 variables.

level1 character vector with the names of the annotation levels.

order numeric vector that indicates the hierarchical order.

createLipidDB

*Customizable lipid DBs creator***Description**

It allows to create easy-customizable lipid DBs for annotation with LipidMS package.

Usage

```
createLipidDB(lipid, chains, chains2)
```

Arguments

lipid	character value indicating the class of lipid. See Details.
chains	character vector indicating the FA chains to be employed
chains2	character vector containing the sphingoid bases to be employed if required.

Details

lipidClass argument needs to be one of the following character values: "Cer", "CerP", "GlcCer", "SM", "Carnitine", "CE", "FA", "HFA", "Sph" (sphingoid bases), "SphP", "MG", "LPA", "LPC", "LPE", "LPG", "LPI", "LPS", "FAHFA", "DG", "PC", "PE", "PG", "PI", "PS", "PA", "TG", "CL", "PCo", "PCp", "PEo", "PEp", "LPCo", "LPCp", "LPEo", "LPEp" or "all".

Value

List with the requested dbs (data frames)

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
fas <- c("8:0", "10:0", "12:0", "14:0", "14:1", "15:0", "16:0", "16:1",
"17:0", "18:0", "18:1", "18:2", "18:3", "18:4", "20:0", "20:1", "20:2",
"20:3", "20:4", "20:5", "22:0", "22:1", "22:2", "22:3", "22:4", "22:5",
"22:6", "24:0", "24:1", "26:0")
sph <- c("16:0", "16:1", "18:0", "18:1")
sn1alkyllipids <- c("16:0", "18:0", "18:1")
sn1vinyllipids <- c("16:0", "18:0")
newdb <- createLipidDB(lipid = "PC", chains = fas, chains2 = sph)
```

crossTables	<i>Cross the original MS1 peaklist with the annotation results</i>
-------------	--

Description

Cross the original MS1 peaklist with the annotation results.

Usage

```
crossTables(msobject, ppm = 5, rttol = 10, dbs)
```

Arguments

msobject	annotated msobject
ppm	mass tolerance in ppm.
rttol	rt tolerance to match peaks in seconds.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .

Value

msobject with an annotatedPeaklist, which is a data frame with 6 columns: mz, RT, int, LipidMSid, adduct and confidence level for the annotation. When multiple IDs are proposed for the same feature, they are sorted based on the annotation level and score.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

dataProcessing	<i>Process mzXML files individually: peakpicking and isotope annotation</i>
----------------	---

Description

Process mzXML files individually: peakpicking and isotope anotation

Usage

```

dataProcessing(
  file,
  acquisitionmode,
  polarity,
  dmzagglom = 15,
  drtagglom = 500,
  drtclust = 100,
  minpeak = c(5, 3),
  drtgap = 10,
  drtminpeak = c(15, 15),
  drtmaxpeak = c(100, 200),
  recurs = 5,
  sb = c(3, 2),
  sn = 2,
  minint = c(1000, 100),
  weight = c(2, 3),
  dmzIso = 5,
  drtIso = 5,
  verbose = TRUE
)

```

Arguments

file	file path.
acquisitionmode	character value: MS, DIA or DDA.
polarity	character value: negative or positive.
dmzagglom	mz tolerance (in ppm) used for partitioning and clustering.
drtagglom	RT window used for partitioning (in seconds).
drtclust	RT window used for clustering (in seconds).
minpeak	minimum number of measurements required for a peak.
drtgap	maximum RT gap length to be filled (in seconds).
drtminpeak	minimum RT width of a peak (in seconds). At least minpeak within the drtmin-peak window are required to define a peak.
drtmaxpeak	maximum RT width of a single peak (in seconds).
recurs	maximum number of peaks within one EIC.
sb	signal-to-base ratio.
sn	signal-to-noise ratio.
minint	minimum intensity of a peak.
weight	weight for assigning measurements to a peak.
dmzIso	mass tolerance for isotope matching.
drtIso	time window for isotope matching.
verbose	print information messages.

Details

It is important that mzXML files are centroided.

This function executes 2 steps: 1) peak-picking based on `enviPick` package and 2) isotope annotation based on `CAMERA` algorithm.

Numeric arguments accept one or two values for MS1 and MS2, respectively.

Value

an `msoject` that contains metadata of the mzXML file, raw data and extracted peaks.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafe.es>

References

Peak-picking algorithm has been imported from `enviPick` R-package: <https://cran.r-project.org/web/packages/enviPick/index>.

Isotope annotation has been adapted from `CAMERA` algorithm: Kuhl C, Tautenhahn R, Boettcher C, Larson TR, Neumann S (2012). "CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets." *Analytical Chemistry*, 84, 283–289. <http://pubs.acs.org/doi/abs/10.1021/ac202450g>.

See Also

[batchdataProcessing](#) and [setmsbatch](#)

Examples

```
## Not run:
msoject <- dataProcessing("input_file.mzXML", acquisitionmode="DIA", polarity,
  dmzagglom = 25, drtagglom = 500, drtclust = 60, minpeak = c(5, 3),
  drtgap = 5, drtminpeak = 20, drtmaxpeak = 100, recurs = 5, sb = c(3, 2),
  sn = 2, minint = c(1000, 100), weight = 2, dmzIso = 10, drtIso = 5)

## End(Not run)
```

ddaFrag

MS/MS scan extraction of a precursor in DDA

Description

This function searches for the closest precursor selected for MS2 in DDA that matches `mz` tolerance and `RT` window of a list of candidates and extracts their fragments.

Usage

```
ddaFrag(candidates, precursors, rawData, ppm)
```

Arguments

<code>candidates</code>	candidates data frame. Output of findCandidates .
<code>precursors</code>	data frame with the whole list of precursors selected for MS2.
<code>rawData</code>	peaklist for MS2 function (MSMS).
<code>ppm</code>	m/z tolerance in ppm.

Details

MS2 scans for a given precursor are searched within a rt window from $\text{minrt} - \text{rttol}/2$ to $\text{maxrt} + \text{rttol}/2$. If the same precursor was selected several times along the peak, the closest scan to the rt at the peak maximum is selected for annotation.

Coelution score for DDA fragments represents their relative intensity within the MS2 scan.

Value

List of data frames with the fragments for each candidate.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

dgdb

DGs database

Description

In silico generated database for common DGs.

Usage

```
data("dgdb")
```

Format

Data frame with 147 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

fadb

FAs database

Description

In silico generated database for common FAs.

Usage

```
data("fadb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

fahfadb

FAHFAs database

Description

In silico generated database for common FAHFAs.

Usage

```
data("fahfadb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

fillpeaksmbatch	<i>Fill peaks from a grouped msbatch</i>
-----------------	--

Description

Use grouping results to target all peaks from the msbatch in each sample and refill intensities at the features table.

Usage

```
fillpeaksmbatch(msbatch)
```

Arguments

msbatch msbatch obtained from the [groupmsbatch](#) function.

Details

Once grouping has been performed, areas are extracted again for each peak and sample based on the peak parameters defined for each feature (mz and tolerance and initial and final RT).

Value

msbatch

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafe.es>

Examples

```
## Not run:  
msbatch <- fillpeaksmbatch(msbatch)  
  
## End(Not run)
```

findCandidates	<i>Search of lipid candidates of a certain class</i>
----------------	--

Description

Search of lipid candidates from a peaklist based on a set of expected adducts.

Usage

```
findCandidates(
  MS1,
  db,
  ppm,
  rt,
  adducts,
  rttol = 3,
  dbs,
  rawData = data.frame(),
  coelCutoff = 0
)
```

Arguments

MS1	peaklist of the MS function. Data frame with 3 columns: mz, RT (in seconds) and int (intensity).
db	database (i.e. pcdB, dgdb, etc.). Data frame with at least 2 columns: Mass (exact mass) and total (total number of carbons and double bound of the FA chains, i.e. "34:1").
ppm	m/z tolerance in ppm.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	character vector containing the expected adducts to search for (i.e. "M+H", "M+Na", "M-H", etc.). See details.
rttol	rt tolerance in seconds to match adducts.
dbs	list of data bases required for the annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be changed. If data bases have been customized using createLipidDB , they also have to be modified here.
rawData	raw scans data. Output of dataProcessing function (MS1\$rawData).
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied.

Details

[findCandidates](#) looks for matches between the m/z of the MS1 peaklist and the expected m/z of the candidates in the database for each adduct. If several adducts are expected, results are combined.

Adducts allowed are contained in adductsTable data frame, which can be modified if required (see [adductsTable](#)).

Value

Data frame with the found candidates. It contains 6 columns: mz, RT, int (from the peaklist data.frame), ppms, cb (total number of carbons and double bounds of the FA chains) and adducts.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

`getInclusionList`*Obtain an inclusion list from the annotation results*

Description

Obtain an inclusion list for the identified lipids.

Usage

```
getInclusionList(df, dbs)
```

Arguments

df	data frame. Output of identification functions (results table from an msobject or feature table from an msbatch).
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .

Value

Data frame with 6 columns: formula, RT, neutral mass, m/z, adduct and the LipidMSid.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

groupmsbatch	<i>Group features from an msbatch</i>
--------------	---------------------------------------

Description

Group features from an msbatch

Usage

```
groupmsbatch(
  msbatch,
  dmz = 5,
  drtagglom = 30,
  drt = 15,
  minsamples,
  minsamplesfrac = 0.25,
  parallel = FALSE,
  ncores,
  deleteduplicates = TRUE,
  thr_overlap_duplicates = 0.7,
  global_gb = getOption("LipidMS.future.globals.maxSizeGB", 24),
  verbose = TRUE
)
```

Arguments

msbatch	msbatch obtained from setmsbatch or alignmsbatch functions.
dmz	mass tolerance between peak groups for grouping in ppm.
drtagglom	rt window for mz partitioning.
drt	rt window for peaks clustering.
minsamples	minimum number of samples represented in clusters used for grouping.
minsamplesfrac	minimum samples fraction represented in each cluster used for grouping. Used to calculate minsamples in case it is missing.
parallel	logical. If TRUE, parallel processing is performed.
ncores	number of cores to be used in case parallel is TRUE.
deleteduplicates	logical. Whether or not duplicated features should be removed after grouping based on the overlap between peak limits. dmz and drt parameters are used to filter the potential duplicates.
thr_overlap_duplicates	numeric value between 0 and 1 to establish the percentage of overlap threshold to consider two features as duplicated.
global_gb	numeric. Gigabytes to set as future.globals.maxSize <i>**inside**</i> the function. Defaults to 'getOption("LipidMS.future.globals.maxSizeGB", 24)'.
verbose	print information messages.

Details

First, peak partitions are created based on the `enviPick` algorithm to speed up the following clustering algorithm. Briefly, peaks are ordered increasingly by `mz` and `RT` and grouped based on user-defined tolerances (`dmz` and `drt`). Each peak is initialized as a partition and then, they are evaluated to decide whether or not they can be joined to the previous partition. If `mz` and `RT` of a peak matches tolerance of any of the peaks in the previous partition, it is reassigned. Then, clustering algorithm is executed to improve these partitions based on their `mz` following the next steps for each partition:

1. Each peak in the partition is initialized as a new cluster. For each cluster we will keep the minimum, maximum and mean value of the `mz`, which at this point have the same values.
2. Calculate a distance matrix between all clusters. This distance will be the greatest difference between minimum and maximum values of each cluster.
3. While any distance is different to NA, search the minimum distance between two clusters.
4. If distance is below the maximum distance allowed, join clusters and update minimum, maximum and mean values, else, set distance to NA and go back to point 3.

Then this same clustering algorithm is executed again to group peaks based on their `RT`. In this case, distances between clusters which share peaks from the same samples will be set to NA.

After groups have been defined, those clusters with a sample representation over `minsamples` or `minsamplesfrac` will be used for building the feature table. Finally, if `deleteduplicates` is set to TRUE, peaks overlap is checked to avoid duplicated or wrongly defined features.

Value

grouped `msbatch`

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafe.es>

References

Partitioning algorithm has been imported from `enviPick` R-package: [https://cran.r-project.org/web/packages/enviPick/index.h](https://cran.r-project.org/web/packages/enviPick/index.html)

Examples

```
## Not run:
msbatch <- groupmsbatch(msbatch)

## End(Not run)
```

hfadb

HFAs database

Description

In silico generated database for common HFAs.

Usage

```
data("hfadb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

idAcylCerneg

Acylceramides (AcylCer) annotation for ESI-

Description

AcylCer identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idAcylCerneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H", "M+CH3COO"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("cbdiff-cer_M-H"),
  chainfrags_sn2 = c("sph_Mn-62.06001", "sph_M-H-H2O"),
  chainfrags_sn3 = c("fa_Mn-1.9918", "fa_Mn-19.0179"),
  intrules = c("cbdiff-cer_sn1/sph_sn2", "sph_sn2/fa_sn3"),
  rates = c("5/1", "2/1"),
  intrequired = c(T, T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for AcylCer in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the sphingoid base. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
chainfrags_sn3	character vector containing the fragmentation rules for the acyl chain. See chainFragments for details.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idAcylCerneg function involves 5 steps. 1) FullMS-based identification of candidate AcylCer as M-H and M+CH₃COO. 2) Search of AcylCer class fragments: no class fragments by default. 3) Search of specific fragments that inform about the acyl chain (Cer as M-H), the sphingoid base (neutral loss of 62.0600 of the Sph) and the FA chain (FA as M-H and M-H₂O but with a N instead of an O, what results in a mass differences of 1.9918 and 19.0179 respectively). 4) Look for possible

chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, the fragment coming from the loss of the acyl chain must be at least 5 times more intense the fragment from the sphingoid base and this one, two times more intense than the FA chain from sn3.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idAcylCerneg(msobject)

## End(Not run)
```

idAcylCerpos

Acylceramides (AcylCer) annotation for ESI+

Description

AcylCer identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idAcylCerpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+H-H2O", "M+Na"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("cbdiff-cer_M+H", "cbdiff-cer_M+H-H2O", "cbdiff-cer_M+H-2H2O"),
  chainfrags_sn2 = c("sph_M+H-H2O", "sph_M+H-2H2O"),
  chainfrags_sn3 = c("fa_Mn+0.02329"),
  intrules = c("sph_sn2/cbdiff-cer_sn1", "sph_sn2/fa_sn3"),
  rates = c("2/1", "5/1"),
  intrequired = c(T, T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for Cer in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the sphingoid base. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.

chainfrags_sn3	character vector containing the fragmentation rules for the acyl chain. See chain-Frags for details.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idAcylCerpos function involves 5 steps. 1) FullMS-based identification of candidate AcylCer as M+H, M+H-H₂O and M+Na. 2) Search of AcylCer class fragments: there are no class fragments by default. 3) Search of specific fragments that inform about the acyl chain (Cer as M+H, M+H-H₂O or M+H-2H₂H), the sphingoid base (Sph as M+H-H₂O or M+H-2H₂O) and the FA chain (FA as M+H but with a N instead of an O, what results in a mass difference of 0.02329 with the Mn of the FA chain). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, Sph fragment must be twice more intense than the loss of the acyl chain and at least 5 times more intense than the FA chain from sn3.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idCerPneg(msobject)

## End(Not run)
```

idBAneg

Bile Acids (BA) annotation for ESI-

Description

BA identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idBAneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  conjfrag = c("baconj_M-H"),
  bafrag = c("ba_M-H-H2O", "ba_M-H-2H2O"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for BA in ESI-. Adducts allowed can be modified in the adductsTable (dbs argument).

conjfrag	character vector containing the fragmentation rules for the BA-conjugates. By default just taurine and glycine are considered, but baconjdb can be modified to add more possible conjugates. See chainFrag s for details. It can also be an empty vector.
bafrag	character vector containing the fragmentation rules for other BA fragments. See chainFrag s for details. It can be an empty vector.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idBAneg function involves 3 steps. 1) FullMS-based identification of candidate BA as M-H. 2) Search of BA-conjugate fragments if required. 3) Search of fragments coming from the loss of H₂O.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (MS-only if no rules are defined, or Subclass level if they are supported by fragments) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idBAneg(msobject)
```

```
## End(Not run)
```

idCarpos

Acylcarnitine annotation for ESI+

Description

Acylcarnitines identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idCarpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(60.0807, 85.0295, "fa_M+H-H2O"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  chainfrags_sn1 = c("fa_M+H-H2O"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for Carnitines in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code>	list of data bases required for annotation. By default, <code>db</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idCarpos` function involves 3 steps. 1) FullMS-based identification of candidate carnitines as M+H and M+Na. 2) Search of carnitine class fragments: 60.0807 and 85.0295 or its loss (FA as M+H-H₂O) coeluting with the precursor ion. 3) Search of specific fragments coming from the FA chain (FA as M+H-H₂O).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as Carnitines only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `msoject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msoject <- idCarpos(msoject)
```

```
## End(Not run)
```

idCEpos

Cholesteryl Esters (CE) annotation for ESI+

Description

CE identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idCEpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("2M+NH4", "2M+Na", "M+NH4", "M+Na"),
  clfrags = c(369.3516, "fa_M+H-H2O"),
  clrequired = c(F, F),
  ftype = c("F", "BB"),
  chainfrags_sn1 = c("fa_M+H-H2O"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for CE in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code>	list of data bases required for annotation. By default, <code>db</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idCEpos` function involves 3 steps. 1) FullMS-based identification of candidate CE as 2M+NH₄, 2M+Na, M+NH₄ and M+Na. 2) Search of CE class fragments: 369.3516 or its loss (FA as M+H-H₂O) coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (FA as M+H-H₂O).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), *m/z*, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (*m/z* error), `confidenceLevel` (in this case, as CE only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `msobject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), *m/z*, RT (in seconds), I (intensity), Adducts, ppm (*m/z* error), `confidenceLevel` (Subclass, FA level, where chains are known but not their positions, or FA position level), `peakID`, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Examples

```
## Not run:
msobject <- idCEpos(msobject)

## End(Not run)
```

idCerneg

*Ceramides (Cer) annotation for ESI-***Description**

Cer identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idCerneg(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H", "M+CH3COO"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("NL-nlsph_M-H", "sph_M-H-2H2O", "sph_M-H-H2O"),
  chainfrags_sn2 = c("fa_Mn-1.9918", "fa_M-H-H2O"),
  intrules = c(),
  rates = c(),
  intrequired = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for Cer in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code>	list of data bases required for annotation. By default, <code>db</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idCerneg` function involves 5 steps. 1) FullMS-based identification of candidate `Cer` as `M-H` and `M+CH3COO`. 2) Search of `Cer` class fragments: there are no class fragment by default. 3) Search of specific fragments that inform about the sphingoid base (`Sph` as `M-H-2H2O` resulting from the loss of the FA chain or loss of part of the sphingoid base) and the FA chain (FA as `M-H` but with a `N` instead of an `O`, what means a mass difference of 1.9918 from the exact mass of the FA or FA as `M-H-H2O`). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default.

Results data frame shows: `ID`, lipid class, `CDB` (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), `mz`, `RT` (in seconds), `I` (intensity, which comes directly from de input), `Adducts`, `ppm` (`mz` error), `confidenceLevel` (Subclass, FA level, where chains are known but not their positions, or FA position level) and `Score` (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `msoject` (list with several elements). The results element is a data frame that shows: `ID`, lipid class, `CDB` (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), `mz`, `RT` (in seconds), `I` (intensity), `Adducts`, `ppm` (`mz` error), `confidenceLevel` (Subclass, FA level, where chains are known but not their positions, or FA position level), `peakID`, and `Score` (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idCerneg(msobject)

## End(Not run)
```

idCerPneg

Ceramides phosphate (CerP) annotation for ESI-

Description

CerP identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idCerPneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(78.9585, 96.9691),
  clrequired = c(F, F),
  ftype = c("F", "F"),
  chainfrags_sn1 = c("sphP_M-H"),
  chainfrags_sn2 = c("fa_Mn-1.9918", ""),
  intrules = c(),
  rates = c(),
  intrequired = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for CerP in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idCerPneg function involves 5 steps. 1) FullMS-based identification of candidate CerP as M-H. 2) Search of CerP class fragments: 78.9585 and 96.9691. 3) Search of specific fragments that inform about the sphingoid base (SphP as M-H resulting from the loss of the FA chain) and the FA chain (FA as M-H but with a N instead of an O, what results in a mass difference of 1.9918 from the exact mass of the FA, or the difference between precursor and sn1 chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idCerPneg(msobject)

## End(Not run)
```

idCerpos

Ceramides (Cer) annotation for ESI+

Description

Ceramides identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idCerpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
```

```

rttol = 3,
rt,
adducts = c("M+H-H2O", "M+Na", "M+H"),
clfrags = c(),
clrequired = c(),
ftype = c(),
chainfrags_sn1 = c("sph_M+H-2H2O"),
chainfrags_sn2 = c(""),
intrules = c(),
rates = c(),
intrequired = c(),
coelCutoff = 0.8,
dbs,
verbose = TRUE
)

```

Arguments

<code>mobject</code>	an <code>mobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for Cer in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.

coelCutoff	coelution score threshold between peaks (adducts, parent and fragment ions...). Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idCerpos function involves 5 steps. 1) FullMS-based identification of candidate Cer as M+H, M+H-H₂O and M+Na. 2) Search of Cer class fragments: there isn't any class fragment by default. 3) Search of specific fragments that inform about the sphingoid base (Sph as M+H-2H₂O resulting from the loss of the FA chain) and the FA chain (by default it is calculated using the difference between precursor and sph fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idCerpos(msobject)

## End(Not run)
```

idCerPpos

*Ceramides phosphate (CerP) annotation for ESI+***Description**

CerP identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idCerPpos(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H"),
  clfrags = c("cer_M+H-H2O", "cer_M+H-2H2O"),
  clrequired = c(F, F),
  ftype = c("BB", "BB"),
  chainfrags_sn1 = c("sph_M+H-2H2O"),
  chainfrags_sn2 = c(""),
  intrules = c(),
  rates = c(),
  intrequired = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for Cer in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFrgs for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
<code>coelCutoff</code>	coelution score threshold between peaks (adducts, parent and fragment ions...). Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code>	list of data bases required for annotation. By default, <code>db</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idCerPpos` function involves 5 steps. 1) FullMS-based identification of candidate CerP as M+H. 2) Search of Cer class fragments: Cer as M+H-H₂O and M+H-2H₂O resulting from the loss of the phosphate group and 1 or 2 H₂O molecules. 3) Search of specific fragments that inform about the sphingoid base (Sph as M+H-2H₂O resulting from the loss of the FA chain and the phosphate group) and the FA chain (by default it is calculated using the difference between precursor and sph fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), `mz`, `RT` (in seconds), `I` (intensity, which comes directly from de input), Adducts, ppm (`mz` error), `confidenceLevel` (Subclass, FA level, where chains are known but not their positions, or FA position level) and `Score` (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `msoject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), `mz`, `RT` (in seconds), `I` (intensity), Adducts, ppm (`mz` error), `confidenceLevel` (Subclass, FA level, where chains are known but not their positions, or FA position level), `peakID`, and `Score` (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idCerPpos(msobject)

## End(Not run)
```

idCLneg

Cardiolipines (CL) annotation for ESI-

Description

CL identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idCLneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 5,
  rt,
  adducts = c("M-H", "M+Na-2H"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("lysopa_M-H-H2O"),
  chainfrags_sn2 = c("lysopa_M-H-H2O"),
  chainfrags_sn3 = c("lysopa_M-H-H2O"),
  chainfrags_sn4 = c("lysopa_M-H-H2O"),
  intrules = c("Unknown"),
  rates = c(),
  intrequired = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msoject	an msoject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for CL in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details.
chainfrags_sn3	character vector containing the fragmentation rules for the chain fragments in sn3 position. See chainFragments for details.
chainfrags_sn4	character vector containing the fragmentation rules for the chain fragments in sn4 position. See chainFragments for details.
intrules	character vector specifying the fragments to compare. See checkIntensityRules . If some intensity rules should be employed to identify the chains position but they are't known yet, use "Unknown". If it isn't required, leave an empty vector.
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idCLneg function involves 5 steps. 1) FullMS-based identification of candidate CL as M-H or M-2H. 2) Search of CL class fragments: no class fragments are searched by defaults as they use to have bad coelution scores. 3) Search of specific fragments that inform about chain composition at sn1 (lysoPA as M-H-H₂O), sn2 (lysoPA as M-H-H₂O), sn3 (lysoPA as M-H-H₂O) and sn4 (lysoPA as M-H-H₂O). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. For CL there are no intensity rules by default.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idCLneg(msobject)

## End(Not run)
```

idDGpos

Diacylglycerols (DG) annotation for ESI+

Description

DG identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idDGpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H-H2O", "M+NH4", "M+Na"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("mg_M+H-H2O"),
  chainfrags_sn2 = c("mg_M+H-H2O"),
  intrules = c("mg_sn1/mg_sn2"),
  rates = c("1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for DG in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .

rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idDGpos function involves 5 steps. 1) FullMS-based identification of candidate DG as M+H-H₂O, M+NH₄ and M+Na. 2) Search of DG class fragments: there are no class fragment by default. 3) Search of specific fragments that inform about the FA chains (MGs as M+H-H₂O resulting from the loss of the FA chains). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position: MG coming from the loss of the sn2 chain is more intense than the one coming from the loss of sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idDGpos(msobject)

## End(Not run)
```

idFAHFAneg

FAHFA annotation for ESI-

Description

FAHFA identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idFAHFAneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("hfa_M-H"),
  chainfrags_sn2 = c("fa_M-H"),
  intrules = c("hfa_sn1/fa_sn2"),
  rates = c("3/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.

adducts	expected adducts for FAHFA in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idFAHFAneg function involves 5 steps. 1) FullMS-based identification of candidate FAHFA as M-H. 2) Search of FAHFA class fragments: there isn't any class fragment by default. 3) Search of specific fragments that inform about chain composition in sn1 (HFA as M-H resulting from the loss of the FA chain) and sn2 (FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, HFA intensity has to be higher than FA.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains

composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idFAHFAneg(msobject)

## End(Not run)
```

idFAneg

Fatty Acids (FA) annotation for ESI-

Description

FA identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idFAneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H", "2M-H"),
  clfrags = c("fa_M-H", "fa_M-H-H2O"),
  clrequired = c(FALSE, FALSE),
  ftype = c("BB", "BB"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>mobject</code>	an <code>mobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for FA in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>db</code> s argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>fctype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code> s	list of data bases required for annotation. By default, <code>db</code> s contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> s may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idFAneg` function involves 2 steps. 1) FullMS-based identification of candidate FA as M-H or 2M-H. 2) Search of FA class fragments: neutral loss of H₂O coeluting with the precursor ion or the molecular ion.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), *m/z*, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (*m/z* error), confidenceLevel (in this case, just MS-only or Subclass level (if any class fragment is defined) are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `mobject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), *m/z*, RT (in seconds), I (intensity), Adducts, ppm (*m/z* error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idFAneg(msobject)

## End(Not run)
```

idLPCneg

Lysophosphocholines (LPC) annotation for ESI-

Description

LPC identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idLPCneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+CH3COO", "M-CH3", "M+CH3COO-CH3"),
  clfrags = c(168.0426, 224.0688, "lysopa_M-H", "lysopc_M-CH3"),
  clrequired = c(F, F, F, F),
  ftype = c("F", "F", "BB", "BB"),
  chainfrags_sn1 = c("fa_M-H"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPC in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPCneg function involves 3 steps. 1) FullMS-based identification of candidate LPC as M+CH₃COO, M-CH₃ and M+CH₃COO-CH₃. To avoid incorrect annotations of PE as PC, candidates which are present just as M-CH₃ will be ignored. 2) Search of LPC class fragments: 168.0426, 224.0688, lysoPA as M-H or lysoPC as M-CH₃ coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (FA as M-H).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPC only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains

composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idLPCneg(msobject)

## End(Not run)
```

idLPCpos

Lysophosphocholines (LPC) annotation for ESI+

Description

LPC identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idLPCpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(104.1075, 184.0739),
  clrequired = c(F, F),
  ftype = c("F", "F"),
  chainfrags_sn1 = c("mg_M+H-H2O"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPC in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPCpos function involves 3 steps. 1) FullMS-based identification of candidate LPC as M+H and M+Na. 2) Search of LPC class fragments: 104.1075 and 184.0739 coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (MG as M+H-H₂O).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPC only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Examples

```
## Not run:
msobject <- idLPCpos(msobject)

## End(Not run)
```

idLPEneg

Lysophosphoethanolamines (LPE) annotation for ESI-

Description

LPE identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idLPEneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(140.0115, 196.038, 214.048, "lysope_M-CH3"),
  clrequired = c(F, F, F, "excluding"),
  ftype = c("F", "F", "F", "BB"),
  chainfrags_sn1 = c("fa_M-H"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPE in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPEneg function involves 3 steps. 1) FullMS-based identification of candidate LPE as M-H. 2) Search of LPE class fragments: 140.0115, 196.038 and 214.048 coeluting with the precursor ion. If a loss of CH₃ group is found coeluting with any candidate, this will be excluded as it is a characteristic fragment of LPC. 3) Search of specific fragments that confirm chain composition (FA as M-H).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPE only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains

composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idLPEneg(msobject)

## End(Not run)
```

idLPEpos

Lysophosphoethanolamines (LPE) annotation for ESI+

Description

LPE identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idLPEpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(141.01909),
  clrequired = c(F),
  ftype = c("NL"),
  chainfrags_sn1 = c("mg_M+H-H2O"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPE in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPEpos function involves 3 steps. 1) FullMS-based identification of candidate LPE as M+H and M+Na. 2) Search of LPE class fragments: neutral loss of 141.01909 coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition in sn1 (MG as M+H-H₂O).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPE only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

Examples

```
## Not run:
msobject <- idLPEpos(msobject)

## End(Not run)
```

idLPGneg

Lysophosphoglycerols (LPG) annotation for ESI-

Description

LPG identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idLPGneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(152.9958, 227.0326, 209.022, 74.0359),
  clrequired = c(F, F, F, F),
  ftype = c("F", "F", "F", "NL"),
  chainfrags_sn1 = c("fa_M-H"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPG in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPGneg function involves 3 steps. 1) FullMS-based identification of candidate LPG as M-H. 2) Search of LPG class fragments: 152.9958, 227.0326, 209.022 and neutral loss of 74.0359 coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (FA as M-H). Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPG only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idLPGneg(msobject)

## End(Not run)
```

idLPIneg

Lysophosphoinositols (LPI) annotation for ESI-

Description

LPI identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idLPIneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(241.0115, 223.0008, 259.0219, 297.0375),
  clrequired = c(F, F, F, F),
  ftype = c("F", "F", "F", "F"),
  chainfrags_sn1 = c("fa_M-H"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPI in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPIneg function involves 3 steps. 1) FullMS-based identification of candidate LPI as M-H. 2) Search of LPI class fragments: 241.0115, 223.0008, 259.0219 and 297.0375 coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (FA as M-H).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPI only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idLPIneg(msobject)

## End(Not run)
```

idLPSneg

Lysophosphoserines (LPS) annotation for ESI-

Description

LPS identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idLPSneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H", "M+Na-2H"),
  clfrags = c(87.032),
  clrequired = c(F),
  ftype = c("NL"),
  chainfrags_sn1 = c("fa_M-H"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for LPS in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments. See chainFrgs for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idLPSneg function involves 3 steps. 1) FullMS-based identification of candidate LPS as M-H and M+Na-2H. 2) Search of LPS class fragments: neutral loss of 87.032 coeluting with the precursor ion. 3) Search of specific fragments that confirm chain composition (FA as M-H).

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as LPS only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idLPSneg(msobject)

## End(Not run)
```

idMGpos

Monoacylglycerol (MG) annotation for ESI+

Description

MG identification based on fragmentation patterns for LC-MS/MS DIA and DDA data acquired in positive mode.

Usage

```
idMGpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H-H2O", "M+NH4", "M+Na"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.

rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for MG in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idMGpos function involves 2 steps. 1) FullMS-based identification of candidate MG as M+H-H₂O, M+NH₄ and M+Na. 2) Search of MG class fragments if any is assigned.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, just MS-only or Subclass level (if any class fragment is defined) are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idMGpos(msobject)

## End(Not run)
```

idNEG

*Lipids annotation for ESI-***Description**

Lipids annotation based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode. This function compiles all functions written for ESI- annotations.

Usage

```
idNEG(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 5,
  coelCutoff = 0.8,
  lipidClasses = c("FA", "FAHFA", "LPC", "LPE", "LPG", "LPI", "LPS", "PC", "PCo", "PCp",
    "PE", "PEo", "PEp", "PG", "PI", "PS", "Sph", "SphP", "Cer", "CerP", "AcylCer", "SM",
    "CL", "BA"),
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 5 seconds.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
lipidClasses	classes of interest to run the identification functions.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules.
verbose	print information messages.

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification); and the annotatedPeaklist element shows the original MS1 peaklist with the annotations on it.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idNEG(msobject)

## End(Not run)
```

idPCneg

Phosphocholines (PC) annotation for ESI-

Description

PC identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPCneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+CH3COO", "M-CH3", "M+CH3COO-CH3"),
  clfrags = c(168.0426, 224.0688, "pc_M-CH3"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  chainfrags_sn1 = c("lysopc_M-CH3"),
  chainfrags_sn2 = c("fa_M-H", "lysopc_M-CH3"),
  intrules = c("lysopc_sn1/lysopc_sn2"),
  rates = c("3/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
```

```

    dbs,
    verbose = TRUE
)

```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PC in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPCneg function involves 5 steps. 1) FullMS-based identification of candidate PC as M+CH₃COO, M-CH₃ or M+CH₃COO-CH₃. To avoid incorrect annotations of PE as PC, candidates which are present just as M-CH₃ will be ignored. 2) Search of PC class fragments: 168.0426, 224.0688 or

loss of CH₃ coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (lysoPC as M-CH₃ resulting from the loss of the FA chain at sn2) and sn2 (lysoPC as M-CH₃ resulting from the loss of sn1 or FA as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPC from sn1 is at least 3 times more intense than lysoPC from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPCneg(msobject)

## End(Not run)
```

idPConeg

Plasmany Phosphocholines (PCo) annotation for ESI-

Description

PCo identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPConeg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+CH3COO", "M-CH3", "M+CH3COO-CH3"),
  clfrags = c(168.0426, 224.0688, "pco_M-CH3"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  chainfrags_sn1 = c("lysopco_M-CH3", "lysopco_M-CH3-H2O"),
  chainfrags_sn2 = c("fa_M-H", "fa_M-CO2-H"),
  intrules = c("lysopco_sn1/fa_sn2"),
  rates = c(1/3),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PCo in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .

rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPConeg function involves 5 steps. 1) FullMS-based identification of candidate PCo as M+CH₃COO, M-CH₃ or M+CH₃COO-CH₃. To avoid incorrect annotations of PEO as PCo, candidates which are present just as M-CH₃ will be ignored. 2) Search of PCo class fragments: 168.0426, 224.0688 or loss of CH₃ coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (LPCo as M-CH₃ and M-CH₃-H₂O resulting from the loss of the FA chain at sn2) and sn2 (FA as M-H and M-CO₂-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, FA fragments from sn2 are at least 3 times more intense than LPCo from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPCneg(msobject)

## End(Not run)
```

idPCopos

*Plasmanyl Phosphocholines (PCo) annotation for ESI+***Description**

PCo identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPCopos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(104.1075, 184.0739, 183.06604),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "NL"),
  chainfrags_sn1 = c("lysopco_M+H", "lysopco_M+H-H2O"),
  chainfrags_sn2 = c("lysopc_M+H", "lysopc_M+H-H2O", ""),
  intrules = c("lysopco_sn1/lysopc_sn2"),
  rates = c("2/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.

adducts	expected adducts for PC in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPCopos function involves 5 steps. 1) FullMS-based identification of candidate PCo as M+H and M+Na. 2) Search of PC class fragments: 104.1075, 184.0739 and neutral loss of 183.06604 coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (LPCo as M+H or M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (LPC as M+H-H₂O resulting from the loss of the FA chain at sn1 or the difference between precursor and sn1 chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, LPCo from sn1 is at least twice more intense than LPC from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPCopos(msobject)

## End(Not run)
```

idPCpneg

Plasmenyl Phosphocholines (PCp) annotation for ESI-

Description

PCp identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPCpneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+CH3COO", "M-CH3", "M+CH3COO-CH3"),
  clfrags = c(168.0426, 224.0688, "pcp_M-CH3"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  chainfrags_sn1 = c("lysopcp_M-CH3", "lysopcp_M-CH3-H2O"),
```

```

chainfrags_sn2 = c("fa_M-H", "fa_M-CO2-H"),
intrules = c("lysopcp_sn1/fa_sn2"),
rates = c(1/3),
intrequired = c(T),
coelCutoff = 0.8,
dbs,
verbose = TRUE
)

```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PCp in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>dbs</code>	list of data bases required for annotation. By default, <code>dbs</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>dbs</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

idPCpneg function involves 5 steps. 1) FullMS-based identification of candidate PCp as M+CH₃COO, M-CH₃ or M+CH₃COO-CH₃. To avoid incorrect annotations of PEp as PCp, candidates which are present just as M-CH₃ will be ignored. 2) Search of PCp class fragments: 168.0426, 224.0688 or loss of CH₃ coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (LPCp as M-CH₃ and M-CH₃-H₂O resulting from the loss of the FA chain at sn2) and sn2 (FA as M-H and M-CO₂-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, FA fragments from sn2 are at least 3 times more intense than LPCp from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPCpneg(msobject)

## End(Not run)
```

idPCpos

*Phosphocholines (PC) annotation for ESI+***Description**

PC identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPCpos(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(104.1075, 184.0739, 183.06604),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "NL"),
  chainfrags_sn1 = c("lysopc_M+H", "lysopc_M+H-H2O"),
  chainfrags_sn2 = c("lysopc_M+H", "lysopc_M+H-H2O", ""),
  intrules = c("lysopc_sn1/lysopc_sn2"),
  rates = c("2/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PC in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPCpos function involves 5 steps. 1) FullMS-based identification of candidate PC as M+H and M+Na. 2) Search of PC class fragments: 104.1075, 184.0739 and neutral loss of 183.06604 coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (lysoPC as M+H or M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (lysoPC as M+H or M+H-H₂O resulting from the loss of the FA chain at sn1 or the difference between precursor and sn1 chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPC from sn1 is at least twice more intense than lysoPC from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPCpos(msobject)

## End(Not run)
```

idPCppos

Plasmenyl Phosphocholines (PCp) annotation for ESI+

Description

PCp identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPCppos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(104.1075, 184.0739, 183.06604),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "NL"),
  chainfrags_sn1 = c("lysopcp_M+H", "lysopcp_M+H-H2O"),
  chainfrags_sn2 = c("lysopc_M+H-H2O", ""),
  intrules = c("lysopcp_sn1/lysopc_sn2"),
  rates = c("1/2"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PC in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrag s for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrag s for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPCppos function involves 5 steps. 1) FullMS-based identification of candidate PC as M+H and M+Na. 2) Search of PC class fragments: 104.1075, 184.0739 and neutral loss of 183.06604 coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (LPCp as M+H or M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (LPC as M+H-H₂O resulting from the loss of the FA chain at sn1 or the difference between precursor and sn1 chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, LPC from sn2 is at least twice more intense than LPCo from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPCppos(msobject)

## End(Not run)
```

idPEneg

Phosphoethanolamines (PE) annotation for ESI-

Description

PE identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPEneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
```

```

    rttol = 5,
    rt,
    adducts = c("M-H"),
    clfrags = c(140.0118, 196.038, 214.048, "pe_M-CH3"),
    clrequired = c(F, F, F, "excluding"),
    ftype = c("F", "F", "F", "BB"),
    chainfrags_sn1 = c("lysope_M-H"),
    chainfrags_sn2 = c("lysope_M-H", "fa_M-H"),
    intrules = c("lysope_sn1/lysope_sn2"),
    rates = c("3/1"),
    intrequired = c(T),
    coelCutoff = 0.8,
    dbs,
    verbose = TRUE
)

```

Arguments

<code>mobject</code>	an <code>mobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PE in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.

coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEneg function involves 5 steps. 1) FullMS-based identification of candidate PE as M-H. 2) Search of PE class fragments: 140.0115, 196.038, 214.048 ion coeluting with the precursor ion. If a loss of CH₃ group is found coeluting with any candidate, this will be excluded as it is a characteristic fragment of PC. 3) Search of specific fragments that inform about chain composition in sn1 (lysoPE as M-H resulting from the loss of the FA chain at sn2) and sn2 (lysoPE as M-H resulting from the loss of the FA chain at sn1 or FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPE from sn1 is at least 3 times more intense than lysoPE from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPEneg(msobject)

## End(Not run)
```

idPEoneg

*Plasmanyl Phosphoethanolamines (PEo) annotation for ESI-***Description**

PEo identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPEoneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 5,
  rt,
  adducts = c("M-H", "M+NaCH3COO"),
  clfrags = c(140.0118, 196.038, 214.048, "peo_M-CH3"),
  clrequired = c(F, F, F, "excluding"),
  ftype = c("F", "F", "F", "BB"),
  chainfrags_sn1 = c("lysopeo_M-H", "lysopeo_M-H-H2O"),
  chainfrags_sn2 = c("fa_M-H"),
  intrules = c("lysopeo_sn1/fa_sn2"),
  rates = c(1/3),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PEo in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.

clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEoneg function involves 5 steps. 1) FullMS-based identification of candidate PEO as M-H and M+NaCH₃COO. 2) Search of PEO class fragments: 140.0115, 196.038, 214.048 ion coeluting with the precursor ion. If a loss of CH₃ group is found coeluting with any candidate, this will be excluded as it is a characteristic fragment of PCo. 3) Search of specific fragments that inform about chain composition in sn1 (lysoPEO as M-H and M-H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, FA fragments from sn2 are at least 3 times more intense than LPEO from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPEoneg(msobject)

## End(Not run)
```

idPEopos

Plasmany Phosphoethanolamines (PEo) annotation for ESI+

Description

PEo identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPEopos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(141.0193),
  clrequired = c(F),
  ftype = c("NL"),
  chainfrags_sn1 = c("lysopeo_M+H", "lysopeo_M+H-H2O"),
  chainfrags_sn2 = c("mg_M+H-H2O"),
  intrules = c("lysopeo_sn1/mg_sn2"),
  rates = c("2/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PE in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEopos function involves 5 steps. 1) FullMS-based identification of candidate PE as M+H and M+Na. 2) Search of PE class fragments: loss of head group (NL of 141.0193) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (LPEo as M+H or M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (MG as M+H-H₂O resulting just from the loss of the head group and the FA chain at sn1). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. LPEo from sn1 is at least 2 times more intense than MG from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPEopos(msobject)

## End(Not run)
```

idPEpneg

Plasmenyl Phosphoethanolamines (PEp) annotation for ESI-

Description

PEp identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPEpneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
```

```

rttol = 5,
rt,
adducts = c("M-H", "M+NaCH3COO"),
clfrags = c(140.0118, 196.038, 214.048, "pep_M-CH3"),
clrequired = c(F, F, F, "excluding"),
ftype = c("F", "F", "F", "BB"),
chainfrags_sn1 = c("lysopep_M-H", "lysopep_M-H-H2O"),
chainfrags_sn2 = c("fa_M-H"),
intrules = c("lysopep_sn1/fa_sn2"),
rates = c(1/3),
intrequired = c(T),
coelCutoff = 0.8,
dbs,
verbose = TRUE
)

```

Arguments

<code>mobject</code>	an <code>mobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PEp in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.

coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEpneg function involves 5 steps. 1) FullMS-based identification of candidate PEp as M-H and M+NaCH₃COO. 2) Search of PEp class fragments: 140.0115, 196.038, 214.048 ion coeluting with the precursor ion. If a loss of CH₃ group is found coeluting with any candidate, this will be excluded as it is a characteristic fragment of PCp. 3) Search of specific fragments that inform about chain composition in sn1 (lysoPEp as M-H and M-H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, FA fragments from sn2 are at least 3 times more intense than LPEp from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPEoneg(msobject)

## End(Not run)
```

idPEpos

*Phosphoethanolamines (PE) annotation for ESI+***Description**

PE identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPEpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c("dg_M+H-H2O"),
  clrequired = c(F),
  ftype = c("BB"),
  chainfrags_sn1 = c("lysope_M+H-H2O", "mg_M+H-H2O"),
  chainfrags_sn2 = c("mg_M+H-H2O"),
  intrules = c("lysope_sn1/lysope_sn1", "mg_sn1/mg_sn2"),
  rates = c("3/1", "1/2"),
  intrequired = c(F, F),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msobject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PE in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.

clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEpos function involves 5 steps. 1) FullMS-based identification of candidate PE as M+H and M+Na. 2) Search of PE class fragments: loss of head group (DG as M+H-H₂O) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (MG as M+H-H₂O resulting from the loss of the FA chain at sn2 and the head group or LPE as M+H-H₂O resulting just from the loss of the FA chain) and sn2 (MG as M+H-H₂O resulting from the loss of the head group and FA chain from sn2). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. LPE or MG from sn1 is at least 3 times more intense than the ones from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPEpos(msobject)

## End(Not run)
```

idPEppos

Plasmenyl Phosphoethanolamines (PEp) annotation for ESI+

Description

PEp identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPEppos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(140.012),
  clrequired = c(F),
  ftype = c("NL"),
  chainfrags_sn1 = c("lysopep_M+H", "lysopep_M+H-H2O"),
  chainfrags_sn2 = c("mg_M+H-H2O"),
  intrules = c("lysopep_sn1/mg_sn2"),
  rates = c("1/3"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PE in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPEppos function involves 5 steps. 1) FullMS-based identification of candidate PE as M+H and M+Na. 2) Search of PE class fragments: loss of head group (NL of 140.012) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (LPEp as M+H or M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (MG as M+H-H₂O from sn2 resulting from the loss of the FA chain at sn1). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. MG from sn2 is at least 3 times more intense than LPEp from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPEppos(msobject)

## End(Not run)
```

idPGneg

Phosphoglycerols (PG) annotation for ESI-

Description

PG identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPGneg(  
  msobject,  
  ppm_precursor = 5,  
  ppm_products = 10,
```

```

    rttol = 3,
    rt,
    adducts = c("M-H"),
    clfrags = c(152.9958, 227.0326, 209.022, 74.0359),
    clrequired = c(F, F, F, F),
    ftype = c("F", "F", "F", "NL"),
    chainfrags_sn1 = c("lysopg_M-H"),
    chainfrags_sn2 = c("lysopg_M-H", "fa_M-H"),
    intrules = c("lysopg_sn1/lysopg_sn2"),
    rates = c("2/1"),
    intrequired = c(T),
    coelCutoff = 0.8,
    dbs,
    verbose = TRUE
)

```

Arguments

<code>mobject</code>	an mobject returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PG in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.

coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPGneg function involves 5 steps. 1) FullMS-based identification of candidate PG as M-H. 2) Search of PG class fragments: 152.9958, 227.0326, 209.022 and neutral loss of 74.0359 coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (lysoPG as M-H resulting from the loss of the FA chain at sn2) and sn2 (lysoPG as M-H resulting from the loss of the FA chain at sn1 or FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPG from sn1 is at least 3 times more intense than lysoPG from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPGneg(msobject)

## End(Not run)
```

idPGpos

*Phosphoglycerols (PG) annotation for ESI+***Description**

PG identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPGpos(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+NH4", "M+Na"),
  clfrags = c("dg_M+H-H2O"),
  clrequired = c(F),
  ftype = c("BB"),
  chainfrags_sn1 = c("mg_M+H-H2O"),
  chainfrags_sn2 = c("mg_M+H-H2O"),
  intrules = c("mg_sn1/mg_sn2"),
  rates = c("1/2"),
  intrequired = c(F),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PE in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>db</code>	list of data bases required for annotation. By default, <code>db</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>db</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

`idPGpos` function involves 5 steps. 1) FullMS-based identification of candidate PG as M+H, M+NH₄ and M+Na. 2) Search of PG class fragments: loss of head group (DG as M+H-H₂O) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (MG as M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (MG as M+H-H₂O resulting from the loss of the FA chain at sn1). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. MG from sn2 is at least twice more intense than the one from sn1.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated `msoject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPGpos(msobject)

## End(Not run)
```

idPIneg

Phosphoinositols (PI) annotation for ESI-

Description

PI identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPIneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(241.0115, 223.0008, 259.0219, 297.0375),
  clrequired = c(F, F, F, F),
  ftype = c("F", "F", "F", "F"),
  chainfrags_sn1 = c("lysopi_M-H", "lysopa_M-H"),
  chainfrags_sn2 = c("lysopi_M-H", "lysopa_M-H", "fa_M-H"),
  intrules = c("lysopi_sn1/lysopi_sn2", "lysopa_sn1/lysopa_sn2"),
  rates = c("3/1", "3/1"),
  intrequired = c(F, F),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

mobject	an mobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PI in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrag s for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrag s for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPIneg function involves 5 steps. 1) FullMS-based identification of candidate PI as M-H. 2) Search of PI class fragments: 241.0115, 223.0008, 259.0219 and 297.0375 coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (lysoPI as M-H resulting from the loss of the FA chain at sn2 or lysoPA as M-H if it also losses the head group) and sn2 (lysoPI or lysoPA as M-H resulting from the loss of the FA chain at sn1 or FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPI or lysoPA from sn1 is at least 3 times more intense than lysoPI or lysoPA from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPIneg(msobject)

## End(Not run)
```

idPIpos

Phosphoinositols (PI) annotation for ESI+

Description

PI identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idPIpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
```

```

rttol = 3,
rt,
adducts = c("M+H", "M+NH4", "M+Na"),
clfrags = c("dg_M+H-H2O"),
clrequired = c(F),
ftype = c("BB"),
chainfrags_sn1 = c("mg_M+H-H2O", "lysopi_M+H-H2O"),
chainfrags_sn2 = c("mg_M+H-H2O", "lysopi_M+H-H2O"),
intrules = c("mg_sn1/mg_sn2", "lysopi_sn1/lysopi_sn2"),
rates = c("2/1", "2/1"),
intrequired = c(F, F),
coelCutoff = 0.8,
dbs,
verbose = TRUE
)

```

Arguments

<code>mobject</code>	an <code>mobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PE in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn1</code> position. See chainFragments for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in <code>sn2</code> position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and <code>sn1</code> chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.

coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPIpos function involves 5 steps. 1) FullMS-based identification of candidate PI as M+H, M+NH₄ and M+Na. 2) Search of PI class fragments: loss of head group (DG as M+H-H₂O) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (MG as M+H-H₂O or LPI as M+H-H₂O resulting from the loss of the FA chain at sn2) and sn2 (MG as M+H-H₂O or LPI as M+H-H₂O resulting from the loss of the FA chain at sn1). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. MG or LPI from sn1 are at least twice more intense than the ones from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPIpos(msobject)

## End(Not run)
```

idPOS

*Lipids annotation for ESI+***Description**

Lipids annotation based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode. This function compiles all functions written for ESI+ annotations.

Usage

```
idPOS(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 5,
  coelCutoff = 0.8,
  lipidClasses = c("MG", "LPC", "LPE", "PC", "PCo", "PCp", "PE", "PEo", "PEp", "PG",
    "PI", "Sph", "SphP", "Cer", "AcylCer", "CerP", "SM", "Carnitines", "CE", "DG", "TG"),
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 5 seconds.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>lipidClasses</code>	classes of interest to run the identification functions.
<code>dbs</code>	list of data bases required for annotation. By default, <code>dbs</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>dbs</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Value

annotated `msobject` (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification); and the `annotatedPeaklist` element shows the original MS1 peaklist with the annotations on it.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idPOS(msobject)

## End(Not run)
```

idPSneg

Phosphoserines (PS) annotation for ESI-

Description

PS identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idPSneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H", "M+Na-2H"),
  clfrags = c(87.032, 152.9958),
  clrequired = c(F, F),
  ftype = c("NL", "F"),
  chainfrags_sn1 = c("lysopa_M-H", "lysopa_M-H-H2O"),
  chainfrags_sn2 = c("lysopa_M-H", "lysopa_M-H-H2O", "fa_M-H"),
  intrules = c("lysopa_sn1/lysopa_sn2"),
  rates = c("3/1"),
  intrequired = c(T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.

rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for PS in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFragments for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFragments for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idPSneg function involves 5 steps. 1) FullMS-based identification of candidate PS as M-H or M+Na-2H. 2) Search of PS class fragments: neutral loss of 87.032 (serine) coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition at sn1 (lysoPA as M-H or M-H-H₂O resulting from the loss of the FA chain at sn2 and the head group) and sn2 (lysoPA as M-H or M-H-H₂O resulting from the loss of the FA chain at sn1 and the head group or FA chain as M-H). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, lysoPA from sn1 is at least 3 times more intense than lysoPA from sn2.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idPSneg(msobject)

## End(Not run)
```

idSMneg

Sphingomyelins (SM) annotation for ESI-

Description

SM identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idSMneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+CH3COO", "M-CH3", "M+CH3COO-CH3"),
  clfrags = c(168.0426, 224.0688, "sm_M-CH3"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  chainfrags_sn1 = c("sph_Mn+150.032"),
```

```

chainfrags_sn2 = c("fa_Mn-1.9918", ""),
intrules = c(),
rates = c(),
intrequired = c(),
coelCutoff = 0.8,
dbs,
verbose = TRUE
)

```

Arguments

<code>msoject</code>	an <code>msoject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for PC in ESI-. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
<code>ftype</code>	character vector indicating the type of fragments in <code>clfrags</code> . It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
<code>chainfrags_sn1</code>	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
<code>chainfrags_sn2</code>	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
<code>intrules</code>	character vector specifying the fragments to compare. See checkIntensityRules .
<code>rates</code>	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
<code>intrequired</code>	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
<code>coelCutoff</code>	coelution score threshold between parent and fragment ions. Only applied if <code>rawData</code> info is supplied. By default, 0.8.
<code>dbs</code>	list of data bases required for annotation. By default, <code>dbs</code> contains the required data frames based on the default fragmentation rules. If these rules are modified, <code>dbs</code> may need to be supplied. See createLipidDB and assignDB .
<code>verbose</code>	print information messages.

Details

idSMneg function involves 5 steps. 1) FullMS-based identification of candidate SM as M+CH₃COO, M-CH₃ or M+CH₃COO-CH₃. 2) Search of SM class fragments: 168.0426, 224.0688 or loss of CH₃ coeluting with the precursor ion. 3) Search of specific fragments that inform about chain composition in sn1 (Sph+phosphocholine as M-CH₃-H₂O which results in a mass difference of Sph+150.032) and sn2 (difference between precursor and sn1 chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idSMneg(msobject)

## End(Not run)
```

idSMpos

*Sphingomyelins (SM) annotation for ESI+***Description**

SM identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idSMpos(
  msubject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H", "M+Na"),
  clfrags = c(104.1075, 184.0739, 183.06604),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "NL"),
  chainfrags_sn1 = c("sph_M+H-2H2O"),
  chainfrags_sn2 = c(""),
  intrules = c(),
  rates = c(),
  intrequired = c(),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

<code>msubject</code>	an <code>msobject</code> returned by dataProcessing .
<code>ppm_precursor</code>	mass tolerance for precursor ions. By default, 5 ppm.
<code>ppm_products</code>	mass tolerance for product ions. By default, 10 ppm.
<code>rttol</code>	total rt window for coelution between precursor and product ions. By default, 3 seconds.
<code>rt</code>	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
<code>adducts</code>	expected adducts for SM in ESI+. Adducts allowed can be modified in <code>adductsTable</code> (<code>dbs</code> argument).
<code>clfrags</code>	vector containing the expected fragments for a given lipid class. See checkClass for details.
<code>clrequired</code>	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.

ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrgs for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrgs for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idSMpos function involves 5 steps. 1) FullMS-based identification of candidate SM as M+H and M+Na. 2) Search of SM class fragments: 104.1075, 184.0739 and neutral loss of 183.06604 coeluting with the precursor ion. 3) Search of specific fragments that inform about the composition of the sphingoid base (Sph as M+H-2H₂O resulting from the loss of the FA chain) and the FA chain (by default it is calculated using the difference between precursor and sph chain fragments). 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In this case, there are no intensity rules by default as FA chain is unlikely to be detected.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated mobject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idSMpos(msobject)

## End(Not run)
```

idSphneg

Sphingoid bases (Sph) annotation for ESI-

Description

Sph identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idSphneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c("sph_M-H-H2O", "sph_M-H-2H2O"),
  clrequired = c(F, F),
  ftype = c("BB", "BB"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.

rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for Sph in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idSphneg function involves 2 steps. 1) FullMS-based identification of candidate Sph as M-H. 2) Search of Sph class fragments: neutral loss of 1 or 2 H₂O molecules.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as Sph only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idSphPneg(msobject)

## End(Not run)
```

idSphPneg

Sphingoid bases phosphate (SphP) annotation for ESI-

Description

SphP identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in negative mode.

Usage

```
idSphPneg(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M-H"),
  clfrags = c(78.9585, 96.9691, "sphP_M-H-H2O"),
  clrequired = c(F, F, F),
  ftype = c("F", "F", "BB"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.

adducts	expected adducts for SphP in ESI-. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idSphpos function involves 2 steps. 1) FullMS-based identification of candidate SphP as M-H. 2) Search of SphP class fragments: 78.9585, 96.969 or neutral loss of 1 H₂O molecule.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as SphP only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maialba@alumni.uv.es>

Examples

```
## Not run:
msobject <- idSphPneg(msobject)

## End(Not run)
```

idSphpos

Sphingoid bases (Sph) annotation for ESI-

Description

Sph identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idSphpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H"),
  clfrags = c("sph_M+H-H2O", "sph_M+H-2H2O"),
  clrequired = c(F, F),
  ftype = c("BB", "BB"),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursors and product ions. By default, 3 seconds.
rt	rt window where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for Sph in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.

clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idSphpos function involves 2 steps. 1) FullMS-based identification of candidate Sph as M+H. 2) Search of Sph class fragments: neutral loss of 1 or 2 H₂O molecules.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as Sph only have one chain, only Subclass and FA level are possible) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idSphpos(msobject)

## End(Not run)
```

idSphPpos

*Sphingoid bases phosphate (SphP) annotation for ESI+***Description**

SphP identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idSphPpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+H"),
  clfrags = c("sphP_M+H-H2O", "sphP_M+H-2H2O", "sphP_M+H-H2O-NH4"),
  clrequired = c(F, F, F),
  ftype = c("BB", "BB", "BB"),
  coelCutoff = 0.7,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursors and product ions. By default, 3 seconds.
rt	rt window where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for Sph in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.
clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.

db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idSphPpos function involves 2 steps. 1) FullMS-based identification of candidate SphP as M+H. 2) Search of SphP class fragments: neutral loss of 1 or 2 H₂O molecules, or H₂O and NH₄.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (in this case, as SphP only have one chain, only Subclass and FA level are possible). and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idSphPpos(msobject)

## End(Not run)
```

idTGpos

*Triacylglycerols (TG) annotation for ESI+***Description**

TG identification based on fragmentation patterns for LC-MS/MS DIA or DDA data acquired in positive mode.

Usage

```
idTGpos(
  msobject,
  ppm_precursor = 5,
  ppm_products = 10,
  rttol = 3,
  rt,
  adducts = c("M+NH4", "M+Na"),
  clfrags = c(),
  clrequired = c(),
  ftype = c(),
  chainfrags_sn1 = c("cbdiff-dg_M+H-H2O"),
  chainfrags_sn2 = c("cbdiff-dg_M+H-H2O"),
  chainfrags_sn3 = c("cbdiff-dg_M+H-H2O"),
  intrules = c("cbdiff-dg_sn2/cbdiff-dg_sn1", "cbdiff-dg_sn2/cbdiff-dg_sn3",
    "cbdiff-dg_sn1/cbdiff-dg_sn3"),
  rates = c("1", "1", "1"),
  intrequired = c(T, T, T),
  coelCutoff = 0.8,
  dbs,
  verbose = TRUE
)
```

Arguments

msobject	an msobject returned by dataProcessing .
ppm_precursor	mass tolerance for precursor ions. By default, 5 ppm.
ppm_products	mass tolerance for product ions. By default, 10 ppm.
rttol	total rt window for coelution between precursor and product ions. By default, 3 seconds.
rt	rt range where the function will look for candidates. By default, it will search within all RT range in MS1.
adducts	expected adducts for TG in ESI+. Adducts allowed can be modified in adductsTable (dbs argument).
clfrags	vector containing the expected fragments for a given lipid class. See checkClass for details.

clrequired	logical vector indicating if each class fragment is required or not. If any of them is required, at least one of them must be present within the coeluting fragments. See checkClass for details.
ftype	character vector indicating the type of fragments in clfrags. It can be: "F" (fragment), "NL" (neutral loss) or "BB" (building block). See checkClass for details.
chainfrags_sn1	character vector containing the fragmentation rules for the chain fragments in sn1 position. See chainFrag s for details.
chainfrags_sn2	character vector containing the fragmentation rules for the chain fragments in sn2 position. See chainFrag s for details. If empty, it will be estimated based on the difference between precursors and sn1 chains.
chainfrags_sn3	character vector containing the fragmentation rules for the chain fragments in sn3 position. See chainFrag s for details. If empty, it will be estimated based on the difference between precursors and sn2 chains.
intrules	character vector specifying the fragments to compare. See checkIntensityRules . If some intensity rules should be employed to identify the chains position but they are't known yet, use "Unknown". If it isn't required, leave an empty vector.
rates	character vector with the expected rates between fragments given as a string (e.g. "3/1"). See checkIntensityRules .
intrequired	logical vector indicating if any of the rules is required. If not, at least one must be verified to confirm the structure.
coelCutoff	coelution score threshold between parent and fragment ions. Only applied if rawData info is supplied. By default, 0.8.
db	list of data bases required for annotation. By default, db contains the required data frames based on the default fragmentation rules. If these rules are modified, db may need to be supplied. See createLipidDB and assignDB .
verbose	print information messages.

Details

idTGpos function involves 5 steps. 1) FullMS-based identification of candidate TG as M+NH₄ and M+Na. 2) Search of TG class fragments: there are no class fragment by default. 3) Search of specific fragments that inform about the FA chains: DGs resulting from the loss of FA chains as M+H-H₂O. 4) Look for possible chains structure based on the combination of chain fragments. 5) Check intensity rules to confirm chains position. In the case of TG, DG resulting from the loss of sn2 if the most intense, followed by the loss of sn1 and sn3, but this FA position level still needs to be improved due to the high level of coelution for TG.

Results data frame shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity, which comes directly from de input), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level) and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Value

annotated msubject (list with several elements). The results element is a data frame that shows: ID, lipid class, CDB (total number of carbons and double bounds), FA composition (specific chains composition if it has been confirmed), mz, RT (in seconds), I (intensity), Adducts, ppm (mz error), confidenceLevel (Subclass, FA level, where chains are known but not their positions, or FA position level), peakID, and Score (parent-fragment coelution score mean in DIA data or relative sum intensity in DDA of all fragments used for the identification).

Note

This function has been written based on fragmentation patterns observed for three different platforms (QTOF 6550 from Agilent, Synapt G2-Si from Waters and Q-exactive from Thermo), but it may need to be customized for other platforms or acquisition settings.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:
msobject <- idTGpos(msobject)

## End(Not run)
```

LipidMSapp

LipidMS shiny app

Description

Interactive UI for LipidMS

Usage

```
LipidMSapp(max_upload_mb = getOption("LipidMS.shiny.maxRequestSizeMB", Inf))
```

Arguments

max_upload_mb max size to upload

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:  
# example data files can be download from github.com/maialba3/LipidMSv2.0_exampleFiles  
  
library(LipidMS)  
LipidMSapp()  
  
## End(Not run)
```

lysopadb*LPAs database*

Description

In silico generated database for common LPAs.

Usage

```
data("lysopadb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

lysopaodb*O-LPA database*

Description

In silico generated database for common O-LPA.

Usage

```
data("lysopaodb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

lysopcdb

LPCs database

Description

In silico generated database for common LPCs.

Usage

```
data("lysopcdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopcdb

O-LPC database

Description

In silico generated database for common O-LPC.

Usage

```
data("lysopcdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopcpdb

P-LPC database

Description

In silico generated database for common P-LPC.

Usage

```
data("lysopcpdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopedb

LPEs database

Description

In silico generated database for common LPEs.

Usage

```
data("lysopedb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopeodb

O-LPE database

Description

In silico generated database for common O-LPE.

Usage

```
data("lysopeodb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopepdb

P-LPE database

Description

In silico generated database for common P-LPE.

Usage

```
data("lysopepdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

lysopgdb*LPGs database*

Description

In silico generated database for common LPGs.

Usage

```
data("lysopgdb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

lysopidb*LPIs database*

Description

In silico generated database for common LPIs.

Usage

```
data("lysopidb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

lysopsdb

LPSs database

Description

In silico generated database for common LPSs

Usage

```
data("lysopsdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

mgdb

MGs database

Description

In silico generated database for common MGs.

Usage

```
data("mgdb")
```

Format

Data frame with 30 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

nlsphdb	<i>Neutral losses db for sphingoid bases. It is employed by idCerneq function.</i>
---------	--

Description

In silico generated database for neutral losses of sphingoid bases in ESI-.

Usage

```
data("nlsphdb")
```

Format

Data frame with 4 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

organizeResults	<i>Prepare output for LipidMS annotation functions</i>
-----------------	--

Description

Prepare a readable output for LipidMS identification functions.

Usage

```
organizeResults(  
  candidates,  
  coelfrags,  
  clfrags,  
  classConf,  
  chainsComb,  
  intrules,  
  intConf,  
  nchains,  
  class,  
  acquisitionmode  
)
```

Arguments

candidates	candidates data frame. Output of findCandidates .
coelfrags	list of coeluting fragments for each candidate
clfrags	vector containing the expected fragments for a given lipid class.
classConf	output of checkClass
chainsComb	output of combineChains
intrules	character vector specifying the fragments to compare. See checkIntensityRules .
intConf	output of checkIntensityRules
nchains	number of chains of the targeted lipid class.
class	character value. Lipid class (i.e. PC, PE, DG, TG, etc.).
acquisitionmode	acquisition mode (DIA or DDA).

Details

Coelution score for DIA data is calculated as the mean coelution score of all fragments used for annotation, while for DDA data, the intensity score is given, which is calculated as the sum of the relative intensities of the fragments used for annotation.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

padb

PAs database

Description

In silico generated database for common PAs.

Usage

```
data("padb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

pcdb*PCs database*

Description

In silico generated database for common PCs.

Usage

```
data("pcdb")
```

Format

Data frame with 147 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

pcodb*O-PC database*

Description

In silico generated database for common O-PC.

Usage

```
data("pcodb")
```

Format

Data frame with 147 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

pcpdb

P-PC database

Description

In silico generated database for common P-PC.

Usage

```
data("pcpdb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

pedb

PEs database

Description

In silico generated database for common PEs.

Usage

```
data("pedb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

peodb

O-PE database

Description

In silico generated database for common O-PE.

Usage

```
data("peodb")
```

Format

Data frame with 30 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

pepdb

P-PE database

Description

In silico generated database for common P-PE.

Usage

```
data("pepdb")
```

Format

Data frame with 147 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

pgdb

PGs database

Description

In silico generated database for common PGs.

Usage

```
data("pgdb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

pidb

PIs database

Description

In silico generated database for common PIs.

Usage

```
data("pidb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

poteicmsbatch	<i>EIC for all samples in a msbatch</i>
---------------	---

Description

EIC for all samples in a msbatch

Usage

```
poteicmsbatch(msbatch, mz, ppm, rt, colorbygroup = TRUE, verbose = TRUE)
```

Arguments

msbatch	msbatch
mz	mz of interest
ppm	mass tolerance in ppm
rt	numeric vector with the RT range to be plotted
colorbygroup	logical. If TRUE, samples will be coloured based on their sample group (from metadata).
verbose	print information messages.

Value

plot

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

plotLipids	<i>Plot informative peaks for lipid annotation</i>
------------	--

Description

Plot informative peaks for each lipid annotated with idPOS and idNEG (or similar functions).

Usage

```
plotLipids(msobject, span = 0.4, ppm = 10, verbose = TRUE)
```

Arguments

msobject	annotated msobject.
span	smoothing parameter. Numeric value between 0 and 1.
ppm	mz tolerance for EIC. If set to 0, the EIC will not be shown.
verbose	print information messages.

Details

Peak intensities are relative to the maximum intensity of each peak to ease visualization.
Grey lines show the the extracted ion chromatograms for the peaks.

Value

msobject with a plots element which contains a list of plots. Plots on the left side represent raw values while plots on the left are smoothed or clean scans (MS2 in DDA).

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

plotticmsbatch	<i>TIC for all samples in a msbatch</i>
----------------	---

Description

TIC for all samples in a msbatch

Usage

```
plotticmsbatch(msbatch, rt, colorbygroup = TRUE)
```

Arguments

msbatch	msbatch
rt	numeric vector with the RT range to be plotted
colorbygroup	logical. If TRUE, samples will be coloured based on their sample group (from metadata).

Value

plot

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

psdb	<i>PSs database</i>
------	---------------------

Description

In silico generated database for common PSs.

Usage

```
data("psdb")
```

Format

Data frame with 147 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

rtdevplot	<i>Plot retention time deviation</i>
-----------	--------------------------------------

Description

Plot retention time deviation of an aligned msbatch

Usage

```
rtdevplot(msbatch, colorbygroup = TRUE)
```

Arguments

msbatch aligned msbatch.

colorbygroup logical. If TRUE, samples will be coloured based on their sample group (from metadata).

Value

plot

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafes.es>

searchIsotopes	<i>Targeted isotopes search</i>
----------------	---------------------------------

Description

This function uses annotation results of deisotoped data to search for isotopes in raw data.

Usage

```
searchIsotopes(  
  msubject,  
  label,  
  adductsTable = LipidMS::adductsTable,  
  ppm = 10,  
  coelCutoff = 0.7,  
  results,  
  dbs  
)
```

Arguments

msubject	msubject.
label	isotope employed for the experiment. It can be "13C" or "D".
adductsTable	adducts table employed for lipids annotation.
ppm	mass error tolerance.
coelCutoff	coelution score threshold between isotopes. By default, 0.7.
results	target list to search isotopes. If missing, all results from the msubject are searched. It is used by searchIsotopesmsbatch .
dbs	list of data bases required for annotation. By default, dbs contains the required data frames based on the default fragmentation rules. If these rules are modified, dbs may need to be supplied. See createLipidDB and assignDB .

Value

List with the isotopes for each compound in the results data frame.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

searchIsotopesmsbatch *Targeted isotopes search for msbatch*

Description

This function uses annotation results of deisotoped data to search for isotopes in raw data.

Usage

```
searchIsotopesmsbatch(  
  msbatch,  
  label,  
  adductsTable = LipidMS::adductsTable,  
  ppm = 10,  
  coelCutoff = 0.7  
)
```

Arguments

msbatch	annotated msbatch.
label	isotope employed for the experiment. It can be "13C" or "D".
adductsTable	adducts table employed for lipids annotation.
ppm	mass error tolerance.
coelCutoff	coelution score threshold between isotopes. By default, 0.7.

Value

List with the isotopes for each compound in the results data frame.

Author(s)

M Isabel Alcoriza-Balaguer <maribel_alcoriza@iislafe.es>

Examples

```
## Not run:  
msbatch <- batchProcessing(metadata = "metadata.csv", polarity = "positive")  
msbatch <- alignmsbatch(msbatch)  
msbatch <- groupmsbatch(msbatch)  
msbatch <- annotatemsbatch(msbatch)  
searchIsotopesmsbatch(msbatch, label = "13C")  
  
## End(Not run)
```

setmsbatch

Create msbatch for batch processing.

Description

Create msbatch from a list of msobjects to build an msbatch.

Usage

```
setmsbatch(msobjectlist, metadata)
```

Arguments

msobjectlist	list of msobjects.
metadata	sample metadata. Optional. It can be a csv file or a data.frame with 3 columns (sample, acquisitionmode and sampletype).

Details

samples are sorted following the metadata data.frame.

Value

msbatch

Author(s)

M Isabel Alcoriza-Balaguer <maialba@iislafes.es>

See Also

[dataProcessing](#) and [batchdataProcessing](#)

Examples

```
## Not run:  
msbatch <- setmsbatch(msobjectlist)  
  
## End(Not run)
```

smdb*SMs database*

Description

In silico generated database for common SMs.

Usage

```
data("smdb")
```

Format

Data frame with 52 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

sphdb*Sphingoid bases database*

Description

In silico generated database for common sphingoid bases.

Usage

```
data("sphdb")
```

Format

Data frame with 4 observations and the following 3 variables.

`formula` character vector containing molecular formulas.

`total` character vector indicating the total number of carbons and double bounds of the chains.

`Mass` numeric vector with the neutral masses.

sphPdb

Sphingoid bases phosphate database

Description

In silico generated database for common sphingoid bases phosphate.

Usage

```
data("sphPdb")
```

Format

Data frame with 4 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

tgdb

TGs database

Description

In silico generated database for common TGs.

Usage

```
data("tgdb")
```

Format

Data frame with 376 observations and the following 3 variables.

formula character vector containing molecular formulas.

total character vector indicating the total number of carbons and double bounds of the chains.

Mass numeric vector with the neutral masses.

Index

* datasets

acylcerdb, [4](#)
adductsTable, [5](#)
baconjdb, [9](#)
badb, [9](#)
carnitinesdb, [12](#)
CEdb, [13](#)
cerdb, [13](#)
cerPdb, [14](#)
cldb, [18](#)
confLevels, [20](#)
dgdb, [25](#)
fadb, [26](#)
fahfadb, [26](#)
hfadb, [31](#)
lysopadb, [135](#)
lysopaodb, [135](#)
lysopcdb, [136](#)
lysopcodb, [136](#)
lysopcpdb, [137](#)
lysopedb, [137](#)
lysopeodb, [138](#)
lysopepdb, [138](#)
lysopgdb, [139](#)
lysopidb, [139](#)
lysopsdb, [140](#)
mgdb, [140](#)
nlsphdb, [141](#)
padb, [142](#)
pcdb, [143](#)
pcodb, [143](#)
pcpdb, [144](#)
pedb, [144](#)
peodb, [145](#)
pepdb, [145](#)
pgdb, [146](#)
pidb, [146](#)
psdb, [149](#)
smdb, [153](#)

sphdb, [153](#)
sphPdb, [154](#)
tgdb, [154](#)

acylcerdb, [4](#)
adductsTable, [5](#), [29](#)
alignmsbatch, [5](#), [30](#)
annotatemsbatch, [7](#)
assignDB, [8](#), [8](#), [22](#), [29](#), [33](#), [36](#), [38](#), [40](#), [42](#), [44](#),
[46](#), [49](#), [51](#), [53](#), [56](#), [58](#), [60](#), [62](#), [64](#), [66](#),
[68](#), [70](#), [72](#), [74](#), [76](#), [79](#), [82](#), [84](#), [86](#), [89](#),
[91](#), [94](#), [96](#), [98](#), [101](#), [103](#), [105](#), [108](#),
[110](#), [112](#), [115](#), [116](#), [118](#), [120](#), [123](#),
[125](#), [127](#), [129](#), [131](#), [133](#), [150](#)

baconjdb, [9](#)
badb, [9](#)
batchdataProcessing, [10](#), [24](#), [152](#)

carnitinesdb, [12](#)
CEdb, [13](#)
cerdb, [13](#)
cerPdb, [14](#)
chainFragments, [14](#), [14](#), [20](#), [33](#), [35](#), [36](#), [38](#), [40](#), [42](#),
[44](#), [46](#), [48](#), [51](#), [53](#), [55](#), [58](#), [62](#), [64](#), [66](#),
[68](#), [70](#), [72](#), [74](#), [79](#), [81](#), [84](#), [86](#), [89](#), [91](#),
[93](#), [96](#), [98](#), [100](#), [103](#), [105](#), [107](#), [110](#),
[112](#), [114](#), [118](#), [120](#), [123](#), [133](#)

checkClass, [15](#), [33](#), [35](#), [39–44](#), [46](#), [48](#), [50](#), [51](#),
[53](#), [55](#), [58](#), [60](#), [62](#), [64](#), [66](#), [68](#), [70](#), [72](#),
[74](#), [76](#), [79](#), [81](#), [84](#), [86](#), [88](#), [89](#), [91](#), [93](#),
[95](#), [96](#), [98](#), [100](#), [102](#), [103](#), [105](#), [107](#),
[109](#), [110](#), [112](#), [114](#), [118](#), [120](#), [122](#),
[123](#), [125](#), [127–130](#), [132](#), [133](#), [142](#)

checkIntensityRules, [17](#), [33](#), [36](#), [44](#), [46](#), [48](#),
[51](#), [53](#), [55](#), [56](#), [58](#), [79](#), [81](#), [82](#), [84](#), [86](#),
[89](#), [91](#), [93](#), [96](#), [98](#), [100](#), [103](#), [105](#),
[107](#), [110](#), [112](#), [114](#), [118](#), [120](#), [123](#),
[133](#), [142](#)

cldb, [18](#)

- coelutingFragments, [14](#), [16](#), [18](#)
coelutionScore, [19](#)
combineChains, [15](#), [17](#), [19](#), [142](#)
confLevels, [20](#)
createLipidDB, [8](#), [14](#), [16](#), [21](#), [22](#), [28](#), [29](#), [33](#),
[36](#), [38](#), [40](#), [42](#), [44](#), [46](#), [49](#), [51](#), [53](#), [56](#),
[58](#), [60](#), [62](#), [64](#), [66](#), [68](#), [70](#), [72](#), [74](#), [76](#),
[79](#), [82](#), [84](#), [86](#), [89](#), [91](#), [94](#), [96](#), [98](#),
[101](#), [103](#), [105](#), [108](#), [110](#), [112](#), [115](#),
[116](#), [118](#), [120](#), [123](#), [125](#), [127](#), [129](#),
[131](#), [133](#), [150](#)
crossTables, [22](#)
dataProcessing, [11](#), [12](#), [18](#), [22](#), [28](#), [33](#), [35](#),
[37](#), [39](#), [41](#), [43](#), [46](#), [48](#), [50](#), [53](#), [55](#), [57](#),
[60](#), [62](#), [64](#), [66](#), [68](#), [70](#), [72](#), [74](#), [75](#), [77](#),
[79](#), [81](#), [83](#), [86](#), [88](#), [91](#), [93](#), [95](#), [98](#),
[100](#), [102](#), [105](#), [107](#), [109](#), [112](#), [114](#),
[116](#), [117](#), [120](#), [122](#), [124](#), [126](#), [128](#),
[130](#), [132](#), [152](#)
ddaFragments, [24](#)
dgdb, [25](#)
fadb, [26](#)
fahfad, [26](#)
fillpeaksbatch, [27](#)
findCandidates, [16](#), [18](#), [20](#), [25](#), [28](#), [29](#), [142](#)
getInclusionList, [29](#)
groupmsbatch, [27](#), [30](#)
hfadb, [31](#)
idAcylCerneg, [32](#)
idAcylCerpos, [34](#)
idBAneg, [37](#)
idCarpos, [39](#)
idCEpos, [41](#)
idCerneg, [43](#)
idCerPneg, [45](#)
idCerpos, [47](#)
idCerPpos, [50](#)
idCLneg, [52](#)
idDGpos, [54](#)
idFAHFAneg, [57](#)
idFAneg, [59](#)
idLPCneg, [61](#)
idLPCpos, [63](#)
idLPEneg, [65](#)
idLPEpos, [67](#)
idLPGneg, [69](#)
idLPIneg, [71](#)
idLPSneg, [73](#)
idMGpos, [75](#)
idNEG, [77](#)
idPCneg, [78](#)
idPConeg, [80](#)
idPCopos, [83](#)
idPCpneg, [85](#)
idPCpos, [88](#)
idPCppos, [90](#)
idPENeg, [92](#)
idPEoneg, [95](#)
idPEopos, [97](#)
idPEpneg, [99](#)
idPEpos, [102](#)
idPEppos, [104](#)
idPGneg, [106](#)
idPGpos, [109](#)
idPIneg, [111](#)
idPIpos, [113](#)
idPOS, [116](#)
idPSneg, [117](#)
idSMneg, [119](#)
idSMpos, [122](#)
idSphneg, [124](#)
idSphPneg, [126](#)
idSphpos, [128](#)
idSphPpos, [130](#)
idTGpos, [132](#)
LipidMSapp, [134](#)
lysopadb, [135](#)
lysopaodb, [135](#)
lysopcdb, [136](#)
lysopcodb, [136](#)
lysopcpdb, [137](#)
lysopedb, [137](#)
lysopeodb, [138](#)
lysopepdb, [138](#)
lysopgdb, [139](#)
lysopidb, [139](#)
lysopsdb, [140](#)
mgdb, [140](#)
nlsphdb, [141](#)
organizeResults, [141](#)

padb, [142](#)
pcdb, [143](#)
pcodb, [143](#)
pcpdb, [144](#)
pedb, [144](#)
peodb, [145](#)
pepdb, [145](#)
pgdb, [146](#)
pidb, [146](#)
ploteicmsbatch, [147](#)
plotLipids, [147](#)
plotticmsbatch, [148](#)
psdb, [149](#)

rtdevplot, [149](#)

searchIsotopes, [150](#)
searchIsotopesmsbatch, [150](#), [151](#)
setmsbatch, [6](#), [11](#), [12](#), [24](#), [30](#), [152](#)
smdb, [153](#)
sphdb, [153](#)
sphPdb, [154](#)

tgdb, [154](#)