Package 'nontarget'

September 27, 2016

Type Package

Title Detecting Isotope, Adduct and Homologue Relations in LC-MS Data

Version 1.9

Date 2016-09-27

Author Martin Loos

Maintainer Martin Loos <mloos@looscomputing.ch>

Description Screening a HRMS data set for peaks related by (1) isotope patterns, (2) different adducts of the same molecule and/or (3) homologue series. The resulting isotopic pattern and adduct groups can then be combined to so-called components, with homologue series information attached. Also allows plotting and filtering HRMS data for mass defects, frequent m/z distances and components vs. non-components.

License GPL-3

Depends enviPat(>= 2.0), nontargetData(>= 1.1), mgcv(>= 1.7-22)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-09-27 16:39:53

R topics documented:

`nontarget-package` *Detecting Isotope, Adduct and Homologue Relations in LC-MS Data.*

Description

Grouping of peaks in a HRMS data set for (1) isotopic pattern relations and (2) different adducts of the same molecule; detection of (3) homologue series. Isotopic pattern and adduct groups can then be related to their (unknown) candidate chemical component, with homologue series information attached. Includes various plotting and filtering functions for e.g. mass defects, frequent m/z distances, components vs. non-components, adduct frequencies.

Details

Package:	nontarget
Type:	Package
Version:	1.9
Date:	2016-03-21
License:	GPL-3

Screens a HRMS data set for peaks related by (1) isotopic patterns and/or (2) different adducts of the same molecule and/or (3) being part of a homologue series, including various plausibility checks. The resulting isotopic pattern groups and adduct groups can then be combined to components, with each component tagged if being part of a homologue series. This does not require prior knowledge about the chemical nature of the components assigned.

Includes various plotting functions, such as (a) m/z vs. RT vs. mass defect, (b) mass defect vs. detected isotope m/z increments, (c) adduct frequencies and their intensity distributions, (d) relations among peaks within single isotope/adduct groups and within single components and (e) homologue series within RT vs. m/z plots. Allows filtering HRMS data for mass defects, satellite peaks, frequent m/z distances and components vs. non-components. Lists of most-common adducts and isotopes are provided or may be user-defined.

Requires HRMS centroid peak lists as input, i.e., a dataframe or matrix with values of (a) m/z, (b) intensity and (c) retention time (RT) per peak. In addition, tolerances for m/z, RT and uncertainties in peak intensity must be defined by the user.

Author(s)

Martin Loos

Maintainer: Martin Loos <Martin.Loos@eawag.ch>

References

Loos, M., Hollender, J., Schymanski, E., Ruff, M., Singer, H., 2012. Bottom-up peak grouping for unknown identification from high-resolution mass spectrometry data. ASMS 2012 annual conference Vancouver, oral session Informatics: Identification.

See Also

Detecting isotope pattern groups: peaklist make.isos pattern.search pattern.search2 plotisotopes plotdefect isotopes resolution_list

Detecting adduct groups: peaklist adduct.search plotadduct adducts

Detecting homologue series: peaklist homol.search plothomol

On combining groups to components: combine plotisotopes plotcomp ms.filter

On filtering and plotting: rm.sat plotall plotgroup ms.filter plotdiff deter.iso

Examples

```
#####################################################
# (0) load required data: ###########################
# (0.1) HRMS peak list & remove satelite peaks: ######
data(peaklist);
peaklist<-rm.sat(peaklist,dmz=0.3,drt=0.1,intrat=0.015,spar=0.8,corcut=-1000,plotit=TRUE);
peaklist<-peaklist[peaklist[,4],1:3];
# (0.2) list of adducts - package enviPat ############
data(adducts);
# (0.3) list of isotopes - package enviPat ###########
data(isotopes);
#####################################################
# (1) run isotope pattern grouping ##################
# (1.1) define isotopes and charge argument ##########
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
# (1.2) run isotope grouping ########################
pattern<-pattern.search(
  peaklist,
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);
# (1.3) plot results ################################
```

```
plotisotopes(pattern);
plotdefect(pattern,elements=c("N"));
#######################################################
# (2.1) run grouping of peaks for different adducts ##
# of the same candidate molecule ####################
adduct<-adduct.search(
  peaklist,
  adducts,
  rttol=0.05,
  mztol=3,
  ppm=TRUE,
  use_adducts=c("M+K","M+H","M+Na","M+NH4"),
  ion_mode="positive"
);
# (2.2) plot results ################################
plotadduct(adduct);
#######################################################
# (3) show single pattern group and its relation #####
# to adduct groups ##################################
plotall(pattern,adduct);
plotgroup(pattern,adduct,groupID=1,massrange=10,allmass=FALSE);
#######################################################
# (4.1) Screen for homologue series #################
homol<-homol.search(
peaklist,
isotopes,
elements=c("C","H","O"),
use_C=TRUE,
minmz=5,
maxmz=120,
minrt=-1,
maxrt=2,
ppm=TRUE,
mztol=3.5,
    rttol=0.5,
minlength=5,
mzfilter=FALSE,
vec_size=3E6,
spar=.45,
R2=.98,
plotit=FALSE
)
# (4.2) Plot results ################################
plothomol(homol,xlim=FALSE,ylim=FALSE,plotlegend=TRUE);
#######################################################
# (5.1) Combine grouping results to components #######
comp<-combine(
pattern,
adduct,
homol,
dont=FALSE,
rules=c(TRUE,FALSE,FALSE)
);
```

```
# (5.2) plot results ################################
plotisotopes(comp);
plotcomp(comp,compoID=1,peakID=FALSE);
####################################################
# (6) Select data from interactive plot ##############
# ms.filter( component=comp,x="mz",y="dm",xlim=FALSE,
# ylim=FALSE,rm.comp=TRUE,plot.comp=TRUE,rm.noncomp=FALSE,
# select.polygon="inside",res=100,filter.for="raw" );
####################################################
```

`adduct.search` *Detecting and grouping adduct m/z relations among peaks in a HRMS dataset*

Description

Algorithm for detecting m/z differences among peaks that may correspond to m/z differences among different adducts.

Usage

```
adduct.search(peaklist, adducts, rttol = 0, mztol = 2,ppm = TRUE,
use_adducts = c("M+H", "M+K", "M+Na"), ion_mode = "positive")
```

Arguments

`peaklist`	Dataframe of HRMS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as `peaklist`.
`adducts`	Data.frame `adducts` or equivalent.
`rttol`	Retention time tolerance. Units as given in column 3 of peaklist argument, e.g. [min].
`mztol`	m/z tolerance setting: value by which the m/z of a peak may vary from its expected value. If parameter `ppm=TRUE` (see below) given in ppm, otherwise, if `ppm=FALSE`, in absolute m/z [u]. Defines the "large" mass tolerance used.
`ppm`	Should `mztol` be set in ppm (TRUE) or in absolute m/z (FALSE)
`use_adducts`	Vector of adducts to be screened for. Corresponds to names in the first column of `adducts`, thus referring to equations from the second column of `adducts` to be used for calculating adduct m/z differences.
`ion_mode`	`"positive"` or `"negative"`.

Details

Given a peak from the peaklist, the `adduct.search` algorithm screens within tolerances `mztol` and `rttol` whether any other peaks may correspond to this one peak via adduct m/z differences. More precisely, the one peak m/z is reset to all possible candidate molecular mass values (M; uncharged, non-adduct). The latter are then used to calculate for all other candidate adduct peaks, which, if found, are subsequently grouped.

For example, consider use_adducts=c("M+H", "M+K"). Given the m/z-value of the one peak, two other peaks with ((m/z*z("M+H")-X("M+H"))/z("M+K"))+X("M+K") and ((m/z*z("M+K")-X("M+K"))/z("M+H"))+X("M+H") are searched for. The peak found for the first term (i.e. with "M+H" being the candidate adduct of the one peak) leads to one group of associated adduct peaks (M+H<->M+K). Another adduct peak (i.e. with "M+K" being the candidate adduct of the one peak) would lead to a second group of associated adduct peaks (M+K<->M+H). Logically, larger adduct groups than the one exemplified can be present, if argument "use_adducts" allows for it (e.g. M+H<->M+K,M+H<->M+Na,M+Na<->M+K).

Value

List of type adduct with 5 entries

`adduct[[1]]`	`Adducts`. Dataframe with peaks (`mass`,`intensity`,`rt`,`peak ID`) and their adduct relations (`to ID`,`adduct(s)`,`mass tolerance`,`charge level`) within adduct groups (`group ID`,`interaction level`).
`adduct[[2]]`	`Parameters`. Parameters used.
`adduct[[3]]`	`Peaks in adduct groups`.Dataframe listing all peaks (`peak IDs`) for an adduct group (`group ID`) and the individual adducts found in that group (`adducts`).
`adduct[[4]]`	`Number of adducts`. Counts of hits per adduct over all adduct groups found.
`adduct[[5]]`	`Overlaps`. Count on how many peaks were assigned to be two different adducts

Note

Peak IDs refer to the order in which peaks are provided. Different IDs exist for adduct groups, isotope pattern groups, grouped homologue series (HS) peaks and homologue series cluster. Yet other IDs exist for the individual components (see note section of `combine`).

The same peak may appear as different adducts in column `adduct[[1]][,7]`, indicating a conflict in assigning the correct adduct. Beware, some adduct combinations from `adducts` may lead to the same results (e.g. M+H<->M+Na vs M+3H<->M+3Na).

Author(s)

Martin Loos

See Also

`rm.sat` `adducts` `peaklist` `plotadduct` `combine` `plotgroup`

Examples

```
#####################################################
# load required data: ###############################
# HRMS peak list: ###################################
data(peaklist)
# list of adducts ###################################
data(adducts)
#####################################################
# run grouping of peaks for different adducts ########
# of the same candidate molecule #####################
adduct<-adduct.search(
  peaklist,
  adducts,
  rttol=0.05,
  mztol=3,
  ppm=TRUE,
  use_adducts=c("M+K","M+H","M+Na","M+NH4"),
  ion_mode="positive"
);
# plot results ######################################
plotadduct(adduct);
#####################################################
```

`combine` *Combining isotope, adduct and homologue series relations in HRMS data sets.*

Description

Combines groups of isotope pattern peaks from `pattern.search` and groups of adduct peaks `adduct.search` to components, with information on homologue series relations from `homol.search` attached. Includes some checks for component plausibility. Needs at least two inputs of (1) isotope pattern relations, (2) adduct relations and (3) homologue series relations. Extracts the most intensive peak per component, allowing for a comparison of components among HRMS data sets.

Individual components and peak relations therein can then be plotted with `plotcomp`. Numbers for detected isotope m/z differences among components can be summarized with `plotisotopes`. Subsets of components and HRMS data can be interactively selected for with `ms.filter`.

Usage

```
combine(pattern, adduct, homol = FALSE, rules = c(FALSE, FALSE, FALSE), dont = FALSE)
```

Arguments

`pattern`	List of type pattern produced by `pattern.search`. If not used, set to `FALSE`.
`adduct`	List of type adduct produced by `adduct.search`. If not used, set to `FALSE`.

homol	List of type homol produced by homol.search. If not used, set to FALSE(default).
rules	Vector with three entries of TRUE or FALSE. See rules section.
dont	Numeric vector with one or several values in between 1 and 4, to exclude components with particular warnings; if not used, set to FALSE. See details.

Details

The algorithm sorts relations among peaks in HRMS data sets generated by pattern.search, adduct.search and homol.search to components in a repetition of four consecutive steps. In a first step, and along decreasing peak intensities, individual peaks are checked for being part of an isotope pattern group and thus relatable to other peaks. In a second step, all peaks within this group from the first step are checked for being part of adduct groups, thus relating to more peaks. Step one and two thereby lead to the set of peaks defining a component. In a third step, all peaks in a component are checked for having adduct or isotope pattern relations to other peaks not yet subsumed into the component, e.g. as a result of overlapping isotope pattern groups. These additional peaks are therefore defined as interfering peaks. In a fourth step, all peaks found for a component are, if available, related to homologue series they may be part of. Once thus assigned to a component, peaks take not further part in subsequent repetitions of step one to initiate a new component (except for interfering peaks, if rules[1]=TRUE). However, they may repeatedly be involved in steps two and three to reflect ambiguities of assigning components.

Four plausibility checks are implemented, represented by warning indices 1 to 4. The first test checks whether the adduct relations found for the peaks assorted under above steps one and two are consistent. If ambiguous adduct relations (e.g. M+H<->M+K AND M+Na<->M+NH4) are found for at least one peak, warning 1 is tagged to the concerned component. The second test checks whether variations in peak intensities within isotope pattern groups are consistent among the different adducts of the same component. This must account for uncertainty in peak intensities via argument inttol of pattern.search. The third check examines whether interfering peaks occur. The fourth check takes effect if a component consists of ambiguously merged isotope pattern groups (only relevant if several charges are used, see use_charges argument in make.isos and the last of the rules in pattern.search). These warning indices can then be used to exclude components affected, using argument dont. For example, dont=c(1,3) excludes components with ambiguous adduct relations and interfering peaks from the final component list.

Value

List of type comp with 7 entries

comp[[1]]	Components. Dataframe with listing of individual components, component IDs and concerned peak IDs and warnings per row. The last columns list m/z, intensity and RT of the most intensive peak in that component.
comp[[2]]	pattern peak list. Entry 1 of list of type pattern produced by pattern.search, i.e. pattern[[1]].
comp[[3]]	adduct peak list. Entry 1 of list of type adduct produced by adduct.search, i.e. adduct[[1]].
comp[[4]]	homologue list. Entry 1 of list of type homol produced by homol.search, i.e. homol[[1]].

`comp[[5]]` Peaks in components. Vector of TRUE orFALSE, indicating if a peak in `pattern[[1]]` or `adduct[[1]]` is part of one or several component(s).

`comp[[6]]` Summary.

`comp[[7]]` Parameters.

Rules setting

`rules[1]`: Set to TRUE enables peaks identified as interfering in a component to enter step one of the algorithm (see details).

`rules[2]`: Set to TRUE to remove single-peaked components.

`rules[3]`: Set to TRUE to only list components being part of (a) homologue serie(s).

Imbecile

Do not combine adduct pattern groups and/or isotope pattern groups and/or homologue series information from (a) different peak lists or (b) the same peak list differently ordered.

Note

Component IDs are allocated in decreasing peak intensity order of the most intensive peak per component, see section value, `comp[[1]]`. In contrast, IDs of individual peaks refer to the order in which peaks are provided.

Setting the argument `pattern` to FALSE skips the first step in the algorithm; adducts group are then only searched for a single peak along decreasing peak intensities. Setting the argument `adduct` to FALSE skips the second step in the algorithm; no adduct groups are then searched for.

Author(s)

Martin Loos

See Also

`pattern.search` `pattern.search2` `adduct.search` `homol.search` `plotisotopes` `plotcomp` `ms.filter` `plotisotopes`

Examples

```
#####################################################
# (0) Group for isotopologues, adducts & homologues  #
data(peaklist);
data(adducts);
data(isotopes);
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
pattern<-pattern.search(
  peaklist,
```

```
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);
adduct<-adduct.search(
  peaklist,
  adducts,
  rttol=0.05,
  mztol=3,
  ppm=TRUE,
  use_adducts=c("M+K","M+H","M+Na","M+NH4"),
  ion_mode="positive"
);
homol<-homol.search(
peaklist,
isotopes,
elements=c("C","H","O"),
use_C=TRUE,
minmz=5,
maxmz=120,
minrt=1,
maxrt=2,
ppm=TRUE,
mztol=3.5,
    rttol=0.5,
minlength=5,
mzfilter=FALSE,
vec_size=3E6,
spar=.45,
R2=.98,
plotit=FALSE
)
#################################################################
# Combine these individual groups to components                 #
#################################################################
# (1) Standard setting:                                         #
# Produce a component list, allowing for single-peaked          #
# components and with interfering peaks also listed as indi-    #
# vidual components (with inputs pattern,adduct,homol):         #
comp<-combine(
pattern,
adduct,
homol,
dont=FALSE,
rules=c(TRUE,FALSE,FALSE)
);
```

```
comp[[6]];
#################################################################
# (2) Produce a list with those components related to a homo-#
# logue series only (requires inputs pattern,adduct,homol):  #
comp<-combine(
pattern,
adduct,
homol,
dont=FALSE,
rules=c(TRUE,FALSE,TRUE)
);
comp[[6]];
#################################################################
# (3) Extract only components that are plausible and contain #
# more than one peak per component, without homologue series #
# information attached (with inputs pattern and adduct):     #
comp<-combine(
pattern,
adduct,
homol=FALSE,
dont=c(1,2,3),
rules=c(TRUE,TRUE,FALSE)
);
comp[[6]];
#################################################################
```

`deter.iso` *Generating list of type iso from filtered m/z differences.*

Description

Produces a list of m/z differences from `diffs` output generated by `plotdiff` to be used as argument `iso` in `pattern.search`. Thus, replaces the `iso` argument to `pattern.search` from `make.isos` and `isotopes` by another `iso` argument of the most frequent m/z differences detected among the HRMS peaks by `plotdiff`.

Usage

```
deter.iso(diffs, histbreaks = 50000, mzmin = 0, mzmax = 0.5,
cutcount = 180, plotit = TRUE)
```

Arguments

`diffs`	vector diffs, i.e. output of function `plotdiff`
`histbreaks`	Number of histogram breaks; thus defines the mztol window to be used with `pattern.search`, see details.
`mzmin`	Minimum value for m/z differences in resulting iso list.

mzmax	Maximum value for m/z differences in resulting iso list.
cutcount	Histogram classes with frequencies below cutcount will be excluded from iso
plotit	Should a plot of the sekected histogram classes be plotted? Recommended.

Details

In addition to returning a iso argument, prints argument values for mztol, ppm and deter of pattern.search when using iso from deter.iso. Also see deter argument of pattern.search.

Value

List of type iso with 5 entries

iso[[1]]	"list of isotopes".
iso[[2]]	"list of isotope masses".
iso[[3]]	"charges".
iso[[4]]	"number of isotope m/z".
iso[[5]]	"elements".

Note

Experimental. For more consistent results, use make.isos instead.

Author(s)

Martin Loos

See Also

pattern.search plotdiff make.isos

Examples

```
data(peaklist)
diffs<-plotdiff(peaklist, histbreaks = 10000, rttol = c(0, 0), mztol = c(0.3, 100), plotit = TRUE);
iso<-deter.iso(diffs,histbreaks=10000,mzmin=0,mzmax=40,cutcount=500,plotit=TRUE);
pattern<-pattern.search(
  peaklist,
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=0.005,
  mzfrac=0.1,
  ppm=FALSE,
  inttol=0.2,
  #rules=c(FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE),
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=TRUE,
  entry=50
```

```
);
```

`homol.search` *Homologue series extraction from LC-MS data.*

Description

Dynamic programming algorithm for unsupervised detection of homologue series in LC-(HR)MS data.

Usage

```
homol.search(peaklist,isotopes,elements=c("C","H","O"),use_C=FALSE,minmz=5,
maxmz=120,minrt=-2,maxrt=2,ppm=TRUE,mztol=3.5,rttol=0.5,minlength=5,
mzfilter=FALSE,vec_size=3E6,mat_size=3,R2=.98,spar=.45,plotit=FALSE,deb=0)
```

Arguments

`peaklist`	Dataframe of picked LC-MS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as `peaklist`.
`isotopes`	Dataframe `isotopes`
`elements`	FALSE or chemical elements in the changing units of the homologue series, e.g. c("C","H") for alkane chains. Used to restrict search.
`use_C`	For `elements`: take element ratio to C-atoms into account? Used to restrict search.
`minmz`	Defines the lower limit of the m/z window to search homologue series peaks, relative to the m/z of the one peak to search from. Absolute m/z value [u].
`maxmz`	Defines the upper limit of the m/z window to search homologue series peaks, relative to the m/z of the one peak to search from. Absolute m/z value [u].
`minrt`	Defines the lower limit of the retention time (RT) window to look for other homologue peaks, relative to the RT of the one peak to search from, i.e., RT+minrt. For decreasing RT with increasing HS mass, use negative values of minrt.
`maxrt`	Defines the upper limit of the RT window to look for other homologue peaks, relative to the RT of the one peak to search from, i.e., RT+maxrt. See `minrt`.
`ppm`	Should `mztol` be set in ppm (`TRUE`) or in absolute m/z [u] (`FALSE`)?
`mztol`	m/z tolerance setting: +/- value by which the m/z of a peak may vary from its expected value. If parameter `ppm=TRUE` (see below) given in ppm, otherwise, if `ppm=FALSE`, in absolute m/z [u].
`rttol`	Retention time (RT) tolerance by which the RT between two adjacent pairs of a homologue series is allowed to differ. Units as given in column 3 of peaklist argument, e.g. [min].
`minlength`	Minimum number of peaks in a homologue series.

mzfilter	Vector of numerics to filter for homologue series with specific m/z differences of their repeating units, given the tolerances in mztol. Mind charge z!
vec_size	Vector size. Ignore unless a relevant error message is printed (then try to increase size).
mat_size	Matrix size for recombining, multiple of input tuples. Ignore unless a relevant error message is printed (then try to increase size).
R2	FALSE or 0<numeric<=1. Coefficient of determination for cubic smoothing spline fits of m/z versus retention time; homologue series with lower R2 are rejected. See smooth.spline.
spar	Smoothing parameter, typically (but not necessarily) in (0,1]. See smooth.spline.
plotit	Logical FALSE or 0<integer<5. Intermediate plots of nearest neigbour paths, spline fits of individual homologues series >=minlength, clustered HS pairs, etc .
deb	Debug returns, ignore.

Details

A dynamic programming approach is used to extract series of peaks that differ in constant m/z units and smooth changes in their retention time within bounds of mass defect changes. First, a nearest neighbour path through a kd-tree representation of the data is used to extract all feasible peak triplets. These triplets are then combined to all plausible n-tupels in n-3 steps. At each such step, each newly formed n-tupel is checked for smooth changes of RT with increasing m/z of the homologues, using cubic splines and a R2-based threshold of the model fit.

Value

List of type homol with 6 entries

homol[[1]]	Homologue Series. Dataframe with peaks (mass,intensity,rt,peak ID) and their homologue series relations (to ID,m/z increment,RT increment) within different homologue series (HS IDs,series level). Last column HS cluster states HS clusters into which a peak was assigned via its HS.
homol[[2]]	Parameters. Parameters used.
homol[[3]]	Peaks in homologue series. Dataframe listing all peaks (peak IDs) per homologue series (HS IDs), the underlying mean m/z & RT increments (m/z increments, RT increments) and the minimum and maximum RT changes between individual peaks of the series.
homol[[4]]	m/z restrictions used. See function argument mzfilter.
homol[[5]]	Peaks per level. List of peak IDs per level in the individual series.
homol[[6]]	Ignore. List with superjacent HS IDs per group - for setdeb=c(3,...)

Warning

The rttol argument of homol.search must not be mixed with that of pattern.search or pattern.search2.

Note

Arguments `isotopes` and `elements` are needed to limit intermediate numbers of m/z differences to screen over, based on feasible changes in mass defect. Similarly, intermediate numbers are also limited by the retention time and m/z windows defined by `minmz/maxmz` and `minrt/maxrt/rttol`, respectively. The latter are always set relative to the individual RT and m/z values of the peaks to be searched from. Overall, these parameters must be chosen carefully to avoid a combinatorial explosion of triplet m/z differences, leading to slow computation, memory problems or senseless results.

Values for `spar` and `R2` have to be adjusted for different chromatographic settings; the smoothing spline fits are used to eliminate homologue series candidates with erratic RT-behaviour. Spline fits at `>=minlength` can be viewed by `plotit=2`.

Peak IDs refer to the order in which peaks are provided. Different IDs exist for adduct groups, isotope pattern groups, grouped homologue series (HS) peaks and homologue series cluster. Yet other IDs exist for the individual components (see note section of `combine`).

Here, IDs of homologue series group are given both in the function output `homol[[1]]`, `homol[[3]]` and `homol[[6]]`, with one homologue series stating one group of interrelated peaks.

Author(s)

Martin Loos

See Also

`rm.sat` `isotopes` `peaklist` `plothomol`

Examples

```
data(peaklist);
data(isotopes)
homol<-homol.search(
peaklist,
isotopes,
elements=c("C","H","O"),
use_C=TRUE,
minmz=5,
maxmz=120,
minrt=-.5,
maxrt=2,
ppm=TRUE,
mztol=3.5,
rttol=0.5,
minlength=5,
mzfilter=FALSE,
vec_size=3E6,
mat_size=3,
spar=.45,
R2=.98,
plotit=FALSE
)
```

```
plothomol(homol);
```

---

make.isos *Deriving list of m/z isotope differences for input into* pattern.search.

Description

Deriving list of m/z isotope differences for input into pattern.search.

Usage

```
make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C",
"15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
```

Arguments

isotopes	Dataframe with isotopes, from dependency enviPat
use_isotopes	Character string of non-monoisotopic isotopes for isotopologue search.
use_charges	Vector of signed integers with length equal to that of use_isotopes. Specifies the charge z for the isotopologue search of each isotope.

Value

List of type iso with 5 entries

iso[[1]]	list of isotopes.
iso[[2]]	list of isotope masses.
iso[[3]]	charges.
iso[[4]]	number of isotope m/z.
iso[[5]]	elements.

Author(s)

Martin Loos

See Also

pattern.search

Examples

```
data(isotopes)
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
```

`ms.filter` *Flexible interactive filtering of components and/or peaks via polygon selection.*

Description

Mark peaks and components in plots of retention time, m/z, mass defect or peak intensity. Select components or peaks by drawing a polygon.

Usage

```
ms.filter(component, x = "mz", y = "dm", xlim = FALSE,
ylim = FALSE, rm.comp = FALSE, plot.comp = TRUE,
rm.noncomp = TRUE, select.polygon = "inside", res = 100,
filter.for = "raw")
```

Arguments

`component`	List of type comp generated by `combine`.
`x`	Scale of x-axis, any of `"mz"` (m/z), `"dm"` (mass defect), `"rt"` (retention time) or `"int"` (intensity).
`y`	Scale of y-axis, any of `"mz"` (m/z), `"dm"` (mass defect), `"rt"` (retention time) or `"int"` (intensity).
`xlim`	xlim=c(upper bound,lower bound), default = `FALSE`.
`ylim`	ylim=c(upper bound,lower bound), default = `FALSE`.
`rm.comp`	Select (i.e. remove) peaks assigned to components by `combine`?
`plot.comp`	Highlight peaks part of a component (red)?
`rm.noncomp`	Select (i.e. remove) peaks not assigned to components by `combine`?
`select.polygon`	Select peaks and/or components to be excluded `"inside"` or `"outside"` of the polygon drawn?
`res`	Resolution of polygon selection; increase if problems with selection by complicated polygons or along polygon boarder occur. Otherwise, ignore.
`filter.for`	What should be filtered and subsequently returned as value by the polygon selection? Any of `"raw"` (raw data, i.e. peak list), `"pattern"` (isotope pattern peak relations, i.e. subset of first entry in list of type pattern generated by `pattern.search`) or `"adduct"` (adduct relations, i.e. subset of first entry in list of type adduct generated by `adduct.search`)

Details

Selection refers to those peaks and/or components to be excluded. If not all peaks in the data set are assigned to components, they are still plotted and can thus e.g. be separated from those assigned to components by setting `rm.comp` vs. `rm.noncomp`.

Value

See `filter.for` argument. Either raw data (i.e. peak list), isotope pattern peak relations, (i.e. subset of first entry in list of type pattern generated by `pattern.search`) or adduct relations (i.e. subset of first entry in list of type adduct generated by `adduct.search`).

Note

Here, mass defect is defined as the difference of m/z to the nearest integer from rounding. `rm.comp = FALSE` and `rm.noncomp = FALSE` leads to no selection and thus no exclusions of anything.

Author(s)

Martin Loos

See Also

`plotcomp` `combine`

`pattern.search` *Detecting and grouping isotope m/z relations among peaks in a HRMS dataset*

Description

Algorithm for detecting isotopes pattern peak groups generated by an unknown candidate chemical component.

Usage

```
pattern.search(peaklist, iso, cutint = min(peaklist[, 2]), rttol = c(-0.5, 0.5),
mztol = 3, mzfrac = 0.1, ppm = TRUE, inttol = 0.5,
rules = c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE),
deter = FALSE, entry = 20)
```

Arguments

`peaklist`	Dataframe of HRMS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as `peaklist`.
`iso`	Object generated by `make.isos` from `isotopes`, defining the isotopes m/z differences to be screened for.
`cutint`	Cutoff intensity. Peaks below this intensity will be (a) omitted and (b) not expected by any of the plausibility rules (see details). See parameter `rules` below.
`rttol`	Minus, plus retention time tolerance. Units as given in column 3 of `peaklist` argument, e.g. [min].
`mztol`	m/z tolerance setting: value by which the m/z of a peak may vary from its expected value. If parameter `ppm=TRUE` (see below) given in ppm, otherwise, if `ppm=FALSE`, in absolute m/z [u]. Defines the "large" mass tolerance used.

mzfrac	"Small" mass tolerance used. Given as a fraction of mztol, see above.
ppm	Should mztol be set in ppm (TRUE) or in absolute m/z (FALSE)
inttol	Intensity tolerance setting: fraction by which peak intensities may vary. E.g. if set to 0.2, a peak with expected intensity 10000 may range in between 8000 and 12000.
rules	Enabling(TRUE)/disabling(FALSE) of rules[1] to rules[11], see details. Vector with eight entries.
deter	If using deter.iso instead of make.isos, set to TRUE. This disables all rules and makes pattern.search compatible with argument iso inputs from deter.iso. Otherwise, ignore.
entry	Memory allocation setting. Increase value if the corresponding warning is issued. Otherwise, ignore.

Details

Detecting groups of isotope pattern peaks involves two steps.

In a first step, and within the given tolerances rttol and mztol, m/z differences among any two peaks are screened for matching differences in m/z among different isotope(s) of an element, as provided by the iso argument. This leads to a set of candidate isotope m/z differences, with each subsequently undergoing four plausibility checks (rules parameter entries 1 to 7).

In a second step, the remaining candidate m/z isotope differences are sorted in tree-like structures (so-called isotope pattern groups), starting from the lowest m/z peak of the data set. Thus, a tree consists of several (>=2) peaks related by isotope m/z differences; the peak with lowest m/z in the tree (root node) represents the monoisotopic peak of the associated candidate molecular component. This does not require prior knowledge about the chemical nature of the components assigned. Again, the resulting trees undergo plausibilization (rules parameter entries 8 to 11).

In addition, groups with m/z isotope differences being detected within "small" mztol are used to calculate a minimum number of atoms per element associated with that m/z isotope difference.

Value

List of type pattern with 12 entries

pattern[[1]]	Patterns. Dataframe with peaks (mass,intensity,rt,peak ID) and their isotope pattern relations (to ID,isotope(s),mass tolerance,charge level) within isotope pattern groups (group ID,interaction level).
pattern[[2]]	Parameters. Parameters used.
pattern[[3]]	Peaks in pattern groups. Dataframe listing all peaks (peak IDs) per isotope pattern group (group ID) at the given z-level(s) (charge level).
pattern[[4]]	Atom counts. Groups with m/z isotope differences being detected within "small" mztol are used to calculate a minimum number of atoms per element associated with that m/z isotope difference.
pattern[[5]]	Count of pattern groups. Number of isotope pattern groups found on the different z-levels used.
pattern[[6]]	Removals by rules. Times rules lead to rejections (rules[1] to rules[10]) or a merging of nested groups (rules[11]).

pattern[[7]]	Number of peaks with pattern group overlapping. Number of overlapping groups; overlap = 1 corresponds to no overlap.
pattern[[8]]	Number of peaks per within-group interaction levels.
pattern[[9]]	Counts of isotopes. Number of times a m/z isotope difference was detected (raw measure / number of isotope pattern groups)
pattern[[10]]	Elements. Elements used via argument iso derived by make.isos.
pattern[[11]]	Charges. z-levels used.
pattern[[12]]	Rule settings. rules[1] to rules[11] settings used.

rules setting

rules[1]: Intensities between two peaks associated via any of the candidate m/z isotope differences of the iso argument are compared. Given this difference in intensity, the minimum number of atoms for the element with highest abundance in argument iso is calculated. If $(minimumnumberofatoms)*(minimummass) > (m/zoflighterpeak * maximumchargeinargumentiso)$, the candidate m/z difference is found implausible and therefore rejected. The minimum mass is set to that of protium (1H) plus its minimum association to numbers of carbon atoms, i.e. 1.0078 + (1/6 * 12.0000). Fast precheck to rules[2] and rules[3].

rules[2]: Repeats rules[1], but uses abundances and minimum masses (including the C-ratios of isotopes) for only those isotope(s) of argument iso ranging within the "large" m/z tolerance set by mztol.

rules[3]: Repeats rules[1], but now uses abundance and minimum masses (including the C-ratios of isotopes) individually for only those isotope(s) of argument iso ranging within the "small" m/z tolerance set by mztol*mzfrac.

rules[4]: If the intensity ratio between two peaks associated via any of the candidate m/z isotope differences of the iso argument is smaller than the smallest isotope abundance ratio of an element of argument iso, the candidate m/z difference is found implausible and therefore rejected. Fast precheck to rules[5] and rules[6].

rules[5]: Repeats rules[4], but now uses abundances for only those isotope(s) of argument iso ranging within the "large" m/z tolerance set by mztol.

rules[6]: Repeats rules[4], but now uses abundances for only those isotope(s) of argument iso ranging within the "small" m/z tolerance set by mztol.

rules[7]: Given those isotopes of argument iso ranging within the "small" m/z tolerance set by mztol and mzfrac and their C-ratio set in isotopes, the minimum number of carbon atoms and the associated 13C peak intensity to be expected at M+1 can be calculated. Checks if this expected 13C peak is present in the data set. If not, the candidate m/z difference is rejected.

rules[8]: Given the intensity and m/z of the monoisotopic peak in a growing isotope pattern tree and values from argument iso, the maximum m/z to which a tree can grow is restrict.

rules[9]: Given (a) the intensities of the monoisotopic peak (=tree root node, interaction level 1) and its first isotopic daughter peaks (tree interaction level 2) and (b) the candidate m/z isotope(s) within the "small" m/z tolerance set by mztol and mzfrac associated with (a), the occurrence of expected peaks (interaction level >2) above the value set by argument cutint is checked. If expected but not found, the peak at interaction level 1 is rejected as being the monoisotopic candidate peak, and a tree is grown on the remaining interrelated peaks. For example, if a monoisotopic peak (= tree interaction level 1) is associated with an intensive 13-C isotope peak (= tree interaction level 2), a

second peak from two 13-C vs. 12-C isotope replacements can be expected and must be checked for.

`rules[10]`: Restriction to `rules[7]` and `[9]`: expected peaks are searched for only if no other measured peaks of higher intensity exist in a tolerance window of absolute m/z = 0.5 around the m/z of the expected peak. This allows skipping the search of expected peaks in cases of intensity masking by other peaks. For example, intensive 37-Cl often mask the occurrence of a second 13-C peak to be expected from `rules[6]`, depending on the number of Cl and C atoms and the measurement resolution used.

`rules[11]`: In some cases, trees may - if several charges z are used - be nested within each other. This rule merges the nested group of charge z=x into the nesting peak group of z>x.

Warning

Acceptable outcomes strongly depend on appropriate parametrization of the algorithm.

Including many isotopes and overly large values for `rttol` and/or `mztol` may lead to overflows. In this case, a warning is issued to increase parameter `entry` or to adjust values of `rttol` and/or `mztol`.

Group IDs are valid both for `pattern[[1]]` and `pattern[[3]]`.

Note

Peak IDs refer to the order in which peaks are provided. Different IDs exist for adduct groups, isotope pattern groups, grouped homologue series (HS) peaks and homologue series cluster. Moreover, and at the highest level, yet other IDs exist for the individual components (see note section of `combine`).

Depending on values of `mztol`, several m/z isotope differences from argument `iso` may match a measured m/z difference between two peaks.

`rules[1]` to `rules[11]` encompass uncertainties in intensity set by parameter `inttol`.

In some cases, two or several isotope pattern trees may overlap. Overlapping trees are not merged by `rules[11]` but only fully nested ones.

Disabling `rules[10]` may in some cases lead to false rejections of candidate m/z isotope differences for `rules[7]` and `rules[9]`, especially for low resolutions.

`rules[9]` is recursive, i.e. may be applied several times on an ever decreasing number of peaks per tree, until plausibility holds or no m/z isotopic differences remain.

Author(s)

Martin Loos

See Also

`pattern.search2` `rm.sat` `peaklist` `make.isos` `plotisotopes` `plotdefect` `combine` `plotgroup isotopes` `resolution_list`

Examples

```
#####################################################
# load required data: ###############################
# HRMS peak list: ###################################
data(peaklist)
peaklist<-rm.sat(peaklist,dmz=0.3,drt=0.1,intrat=0.015,spar=0.8,corcut=-1000,plotit=TRUE);
peaklist<-peaklist[peaklist[,4],1:3];
# list of isotopes ##################################
data(isotopes)
#####################################################
# (1) run isotope pattern grouping ##################
# (1.1) define isotopes and charge (z) argument ######
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
# (1.2) run isotope grouping ########################
# save the list returned as "pattern" ###############
pattern<-pattern.search(
  peaklist,
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);
names(pattern);
# extract peaks listed in isotope pattern group no.1 #
# under pattern[[3]] from pattern[[1]] ##############
pattern[[1]][as.numeric(strsplit(as.character(pattern[[3]][1,2]),",")[[1]]),];
# (1.3) plot results ################################
plotisotopes(pattern);
plotdefect(pattern,elements=c("N"));
#####################################################
```

`pattern.search2` *Detecting and grouping isotope m/z relations among LC-HRMS centroid peaks, based on quantized reference data*

Description

Algorithm for grouping isotope pattern centroids of chemical components by querying quantized simulation data

Usage

```
pattern.search2(peaklist,quantiz,mztol=2,ppm=TRUE,inttol=0.5,rttol=0.3,
use_isotopes=c("13C","37Cl","15N","81Br","34S","18O"),use_charges=c(1,2),
use_marker=TRUE,quick=FALSE,isotopes)
```

Arguments

peaklist	Dataframe of HRMS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as peaklist.
quantiz	Quantized simulation data of feasible centroid-centroid relations as provided by package nontargetData.
mztol	m/z tolerance setting: value by which the m/z of a peak may vary from its expected value. If parameter ppm=TRUE (see below) given in ppm, otherwise, if ppm=FALSE, in absolute m/z [u].
ppm	Should mztol be set in ppm (TRUE) or in absolute m/z (FALSE).
inttol	Intensity tolerance setting = fraction by which peak intensities may vary; e.g., if set to 0.2, a peak with expected intensity 10000 may range in between 8000 and 12000.
rttol	+/- retention time tolerance. Units as given in column 3 of peaklist argument, e.g. [min].
use_isotopes	Restrict query to certain isotopes dominating centroid relations; set to FALSE to use all available isotopes.
use_charges	Vector of signed integers. Restrict query to certain charges z; set to FALSE to use all charge states.
use_marker	Query for marker peaks, FALSE or TRUE?
quick	Continue if query finds first hit? Speeds up, but leaves resulting information on underlying isotopes incomplete.
isotopes	Dataframe of relevant isotopes as provided by package enviPat; used for checking user inputs.

Details

As alternative to rule-based pattern.search, differences among measured centroids (peaklist) are queried to match those of compressed (=quantized) simulation data within bounds of measurement tolerances and the quantization distortion. Hence, in comparion to pattern.search, this approach accounts for centroid mass shifts induced by peak profile interferences prevalent at even high m/z resolution.

To derive the quantized data, isotope pattern centroids of several million organic molecular formulas from the PubChem database were calculated for various classes of adducts. Molecular formulas were filtered to be unique and only to contain C, H, O, N, Cl, Br, K, Na, S, Si, F, P and/or I. The resulting >250 million centroid pairs from individual patterns were then categorized for their dominant isotopologues, charge and the possible presence of another centroid of higher intensity than that of the pair (=marker peak). Within these categories, data on centroid pair (a) m/z, (b) m/z differences, (c) intensity ratios and (d) marker m/z was quantized by a recursive partitioning

procedure. The resulting compressed data representation was extended by nearest neigbour estimates in the above dimensions (a) to (d) to account for queries with molecular formulas possibly not present in the PubChem set. Internally, the quantized simulation data is queried by a tree-like space-partitioning structure for hyperrectangles, while centroids from `peaklist` are restructured into kd-trees.

Value

List of type pattern with 12 entries

`pattern[[1]]` `Patterns`. Dataframe with peaks (`mass`,`intensity`,`rt`,`peak ID`) and their isotope pattern relations (`to ID`,`isotope(s)`,`mass tolerance` (deprecated),`charge level`) within isotope pattern groups (`group ID`,`interaction level` (deprecated)).

`pattern[[2]]` `Parameters`. Parameters used.

`pattern[[3]]` `Peaks in pattern groups`. Dataframe listing all peaks (`peak IDs`) per isotope pattern group (`group ID`) at the given z-level(s) (`charge level`).

`pattern[[4]]` `Atom counts`. Deprecated.

`pattern[[5]]` `Count of pattern groups`. Number of isotope pattern groups found on the different z-levels used.

`pattern[[6]]` `Removals by rules`. Deprecated.

`pattern[[7]]` `Number of peaks with pattern group overlapping`. Deprecated

`pattern[[8]]` `Number of peaks per within-group interaction levels`.

`pattern[[9]]` `Counts of isotopes`. Number of times a m/z isotope difference was detected (raw measure / number of isotope pattern groups)

`pattern[[10]]` `Elements`. Elements used via argument iso derived by `make.isos`.

`pattern[[11]]` `Charges`. z-levels used.

`pattern[[12]]` `Rule settings`. Deprecated.

Warning

Acceptable outcomes strongly depend on appropriate parametrization of the algorithm and using the correct `quantiz` data set from package `nontargetData`. Using overly large values for `rttol` and/or `mztol` may lead to slow execution.

Note

Peak IDs refer to the order in which peaks are provided.

If you do not find quantized simulation data for your instrument in package `nontargetData` and you can provide resolution=f(m/z) information: contact maintainer.

Author(s)

Martin Loos

See Also

`rm.sat` `peaklist` `plotisotopes` `plotdefect` `combine` `plotgroup` `pattern.search`

Examples

```
#####################################################
# load HRMS centroid list: ##########################
data(peaklist)
# load isotope data #################################
data(isotopes)
# load quantized simulation data ####################
data(OrbitrapXL_VelosPro_R60000at400_q)
#####################################################
# run isotope pattern grouping ######################
# save the list returned as "pattern" ###############
pattern<-pattern.search2(
peaklist,
OrbitrapXL_VelosPro_R60000at400_q,
mztol=2,
ppm=TRUE,
inttol=0.5,
rttol=0.3,
use_isotopes=FALSE,
use_charges=FALSE,
use_marker=TRUE,
quick=FALSE,
isotopes
)
names(pattern);
#####################################################
```

`peaklist` *HRMS peak list*

Description

LC-HRMS peak list of a sewage treatment plant effluent sample.

Usage

```
data(peaklist)
```

Format

Numeric data frame with 11172 observations on the following 3 variables.

`mass` Peak m/z

`intensity` Peak intensity

`rt` Peak retention time [min]

Details

HPLC-ESI-FTMS Thermo Fisher orbitrap, resolution 100.000@400m/z, positive ionization, profile data. Generated by Thermo Fisher Formulator peak-picking algorithm; contains satellite peaks for some high intensity peaks.

Examples

```
data(peaklist)
plot(peaklist[,3],peaklist[,1],pch=19,cex=0.5,xlab="Retention time [min]",ylab="m/z")
```

`plotadduct` *Plot of frequencies and peak intensities of different adducts*

Description

Plots absolute frequencies of different adducts detected by `adduct.search` and boxplots the intensity distributions of associated peaks.

Usage

```
plotadduct(adduct)
```

Arguments

`adduct`	List of type adduct produced by `adduct.search`.

Author(s)

Martin Loos

See Also

`adduct.search`

`plotall` *RT vs. m/z scatterplot marking isotope pattern and adduct group peaks.*

Description

RT vs. m/z scatterplot marking isotope pattern and adduct group peaks.

Usage

```
plotall(pattern, adduct)
```

Arguments

pattern	List of type pattern produced by pattern.search.
adduct	List of type adduct produced by adduct.search.

Author(s)

Martin Loos

See Also

pattern.search adduct.search

Examples

```
data(peaklist);
data(adducts);
data(isotopes);
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))
pattern<-pattern.search(
  peaklist,
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);
adduct<-adduct.search(
  peaklist,
  adducts,
  rttol=0.05,
  mztol=3,
  ppm=TRUE,
  use_adducts=c("M+K","M+H","M+Na","M+NH4"),
  ion_mode="positive"
);
plotall(pattern, adduct)
```

plotcomp *Plot and print isotope and adduct relations among peaks of a single component*

Description

Plot and print isotope and adduct relations among peaks of a single component. Also lists all other peaks of the data set within tolerance ranges of m/z and retention time (RT).

Usage

```
plotcomp(comp, compoID, peakID = FALSE)
```

Arguments

comp	List of type comp produced by function combine.
compoID	ID of component to be plotted. For description of component IDs see combine, note section. Use with argument peakID=FALSE.
peakID	ID of a peak in a component; selects the component containing the peak with this ID. For description of peak IDs see note section. Use with argument compoID=FALSE.

Details

The upper plot panel provides a circular plot of peak relations, with m/z increasing clockwise starting from noon. Herein, peaks are represented by their peak IDs; numbers in brackets give decreasing peak intensity ranks over all peaks in the shown component. Adduct relations are symbolized by red lines and isotope relations by blue arrows. Thin instead of thick lines stand for interfering peaks. In addition, all relations, other peaks within range and homologue series information are printed as value of plotcomp

The lower panel barplot shows intensities vs. m/z of both the peaks in the component (bold) and the peaks within tolerance ranges of m/z and RT (grey), defined by arguments mztol and rttol of pattern.search and adduct.search.

Note

Input peaklist is internally sorted and saved in the lists returned by (a) increasing retention time and (b) m/z by all pattern.search, adduct.search and homol.search. Peak IDs refer to this very order - in contrast to group IDs. Different IDs exist for adduct groups, isotope pattern groups, grouped homologue series (HS) peaks and homologue series cluster. Moreover, and at the highest level, IDs exist for the individual components (see note section of combine).

Author(s)

Martin Loos

See Also

combine

plotdefect *Mass defect vs. m/z scatterplot of HRMS peaks, with specific m/z isotope differences highlighted.*

Description

Mass defect vs. m/z scatterplot of HRMS peaks, with specific m/z isotope differences highlighted.

Usage

```
plotdefect(pattern, elements = c("Br"))
```

Arguments

pattern	List of type pattern produced by pattern.search.
elements	Character string of an element for which the isotope m/z differences between two peaks detected by pattern.search should be highlighted (red).

Note

Here, mass defect is defined as the difference of m/z to the nearest integer from rounding. rm.comp = FALSE and rm.noncomp = FALSE leads to no selection and thus no exclusion of anything.

Author(s)

Martin Loos

See Also

pattern.search

Examples

```
data(peaklist);

peaklist<-rm.sat(peaklist,dmz=0.3,drt=0.1,intrat=0.015,spar=0.8,corcut=-1000,plotit=TRUE);

peaklist<-peaklist[peaklist[,4],1:3];

data(isotopes);

iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))

pattern<-pattern.search(
  peaklist,
  iso,
```

```
  cutint=10000,
  rttol=c(-0.05,0.05),
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);

plotdefect(pattern,elements=c("N"));
plotdefect(pattern,elements=c("Cl"));
plotdefect(pattern,elements=c("Br"));
plotdefect(pattern,elements=c("S"));
plotdefect(pattern,elements=c("C"));
plotdefect(pattern,elements=c("K"));
# P has only one isotope, hence:
# plotdefect(pattern,elements=c("P"));
```

`plotdiff` *Filtering important m/z differences among peaks of a HRMS data set.*

Description

Produce a vector and histogram of m/z differences among peaks in a HRMS data set. Frequent m/z differences may be relatable to isotope patterns and the presence of different adducts.

Usage

```
plotdiff(peaklist, histbreaks = 10000, rttol = c(0, 0), mztol = c(0, 100), plotit = TRUE)
```

Arguments

`peaklist`	Dataframe of HRMS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as `peaklist`.
`histbreaks`	Number of histogram breaks.
`rttol`	Window (upper and lower difference bound, relative to the one peak) of retentiom time (RT) differences of peaks to the one peak screened from, see details and note. Units as given in column 3 of peaklist argument, e.g. [min].
`mztol`	Window (upper and lower difference bound, relative to the one peak) of m/z differences [u] of peaks to the one peak screened from, see details.
`plotit`	Should histogram be plotted? If `FALSE`, `plotdiff` will only return a vector of type `diff`, see value.

Details

For each one peak in the dataset, plotdiff screens for other peaks within arguments rttol and mztol, saves their m/z difference to the m/z value of the one peak and, over all one peaks, finally generates a histogram of all these m/z differences. Thus, and depending on the resolution set by argument histbreaks, frequent m/z differences can be visualized.

Value

Vector diffs of m/z differences. Can serve as input to deter.iso.

Note

Argument rttol can e.g. be used to only include m/z differences of peaks with a higher RT relative to that of the one peak (as is the case in homologue series). For example, let one peak have RT=12 min. Using rttol=c(1,4), only m/z differences with peaks having a RT in between 13 and 14 min will then be screened for this one peak. Akin for argument mztol.

Author(s)

Martin Loos

See Also

peaklist

Examples

```
data(peaklist)
diffs<-plotdiff(peaklist, histbreaks = 10000, rttol = c(0, 0), mztol = c(0, 100), plotit = TRUE)
```

---

plotgroup *Barplot of m/z isotope (and optional adduct) relations within an isotope pattern group.*

Description

Plots the m/z isotope relations among peaks part of an isotope pattern group detected by pattern.search. Optionally, adduct relations from adduct.search can be depicted, too.

Usage

```
plotgroup(pattern, adduct = FALSE, groupID, massrange = 10, allmass = TRUE)
```

Arguments

pattern	List of type pattern, i.e. value generated by pattern.search.
adduct	List of type pattern, i.e. value generated by pattern.search. If not used, set to FALSE.
groupID	Isotope pattern group ID of the isotope pattern group to be plotted. Group ID as generated by pattern.search.
massrange	m/z range of other peaks in the HRMS peaks in the data set below and above the smallest and largest m/z of the isotope pattern group to be plotted, respectively.
allmass	Prints only the peaks in the isotope pattern (allmass=FALSE) or also those included via argument massrange (allmass=TRUE).

Details

The upper pannel barplot shows all peaks included by massrange. The lower one only those of the isotope pattern group specified by argument groupID. Below that, shown by lines, come the isotope relations among peaks. At the bottom, relations of individual peaks in the isotope pattern group to adduct groups are highlighted, as far as available. Herein, this adduct refers to the adduct assigned to the isotope pattern group, whereas further adducts to those of other peaks relatable via adduct groups. Peak number refers to the line number of the peak dataframe printed (see value).

Value

Dataframe with peaks, see argument allmass.

Author(s)

Martin Loos

See Also

pattern.search adduct.search

Examples

```
############################################################
data(peaklist);
data(adducts);
data(isotopes);
# run isotope grouping ###################################
iso<-make.isos(isotopes,
use_isotopes=c("13C","15N","34S","37Cl","81Br","41K","13C","15N","34S","37Cl","81Br","41K"),
use_charges=c(1,1,1,1,1,1,2,2,2,2,2,2))

pattern<-pattern.search(
  peaklist,
  iso,
  cutint=10000,
  rttol=c(-0.05,0.05),
```

```
  mztol=2,
  mzfrac=0.1,
  ppm=TRUE,
  inttol=0.2,
  rules=c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),
  deter=FALSE,
  entry=50
);
plotgroup(pattern,adduct=FALSE,groupID=3,massrange=10,allmass=FALSE)
# run adduct grouping ###################################
adduct<-adduct.search(
  peaklist,
  adducts,
  rttol=0.05,
  mztol=3,
  ppm=TRUE,
  use_adducts=c("M+K","M+H","M+Na","M+NH4"),
  ion_mode="positive"
);
plotgroup(pattern,adduct,groupID=3,massrange=10,allmass=FALSE)
#########################################################
```

`plothomol` *Marks homologue series peaks in a scatterplot of retention time (RT) vs. m/z*

Description

Given results from `homol.search`, a scatterplot of peaks within m/z and RT is generated with homologue series marked. Herein, homologue series receive a color code based on the mean m/z differences between adjacent peaks of a series; these differences are rounded up to the second digit.

Usage

```
plothomol(homol, xlim = FALSE, ylim = FALSE,plotlegend=TRUE,plotdefect=FALSE)
```

Arguments

`homol`	List of type homol produed by `homol.search`.
`xlim`	`xlim=c(upper bound,lower bound)`, default = FALSE.
`ylim`	`ylim=c(upper bound,lower bound)`, default = FALSE.
`plotlegend`	Should a listing of m/z differences within homologue series and the concommittant color codes been added to the plot? If not, set to FALSE.
`plotdefect`	Plot the mass defect instead of the m/z value.

Author(s)

Martin Loos

See Also

homol.search

Examples

```
data(peaklist);
data(isotopes)
homol<-homol.search(
peaklist,
isotopes,
elements=c("C","H","O"),
use_C=TRUE,
minmz=5,
maxmz=120,
minrt=2,
maxrt=2,
ppm=TRUE,
mztol=3.5,
    rttol=0.5,
minlength=5,
mzfilter=FALSE,
vec_size=3E6,
spar=.45,
R2=.98,
plotit=FALSE
)
plothomol(homol,xlim=FALSE,ylim=FALSE,plotlegend=FALSE,plotdefect=FALSE);
```

---

plotisotopes *Plot of isotope counts over isotope pattern groups or components.*

Description

Plots and prints counts of m/z isotope differences detected either by pattern.search or by combine.

Usage

```
plotisotopes(input)
```

Arguments

input	Either list of type pattern produced by pattern.search or a list of type comp produced by combine

Details

The function allows to track the number of m/z isotope differences (a) over individual pairs of peaks and (b) aggregated over isotope pattern groups (argument `pattern`) or (c) aggregated over components and (d) aggregated over components within small mass tolerance (argument `comp`). The small mass tolerance is defined by the `massfrac` and `mztol` arguments of `pattern.search` and `adduct.search`.

Value

Dataframe listing counts

Author(s)

Martin Loos

See Also

`pattern.search` `combine`

`rm.sat` *Removal of satellite peaks from FT-MS peak lists*

Description

Brute force method to remove satellite peak from a FT-HRMS peak list.

Usage

```
rm.sat(peaklist, dmz = 0.3, drt = 0.3, intrat = 0.01, spar = 0.8,
corcut = 0.8, plotit = TRUE)
```

Arguments

`peaklist`	Dataframe of HRMS peaks with three numeric columns for (a) m/z, (b) intensity and (c) retention time, such as `peaklist`.
`dmz`	m/z window around a parent peak within which satellite peaks are searched for.
`drt`	Retention time window around a parent peak within which satellite peaks are searched for.
`intrat`	Intensity ratio between satellite peak/associated parent peak below which the former are removed.
`spar`	`spar` argument used in R-function smooth spline. See details and `?smooth.spline`.
`corcut`	Correlation coefficient above which symmetrical peaks are marked as satellite peaks. To disable, set to `-1000` See details.
`plotit`	Plot results?

Details

"Parent" peak refers to a peak having associated satellite peaks as artifacts from FT calculations.

`rm.sat` screens, along decreasing intensity, peaks for having other peaks within ranges set by arguments `dmz`, `drt` and `intrat`. If present, the latter are marked as satellite peaks and are subsequently excluded from further screening within `rm.sat`.

In addition, arguments `spar` and `corcut` evaluate the symmetry of satellite peaks around the parent peak (i.e. below and above the parent peak m/z), if enough peaks around a parent peak within ranges set by arguments `dmz`, `drt` and `intrat` are found (here: at least 8 peaks, 4 above and 4 below the parent peak m/z). Two splines are fitted by R function `smooth.spline`, one to those peaks above and one to those peaks below the parent peak m/z. If the splines are symmetric (i.e. correlated with each other, see argument corcut), the associated peaks are termed satellites. This approach has not yet faced validation and is highly dependent on the peak-picking algorithm.

Value

A dataframe with four columns. The first three columns are identical to those of argument `peaklist`. The fourth columns marks potential satellite peaks with `FALSE`, the other peaks with `TRUE` (see example).

Note

Not removing satellite peaks may lead to undesirable artifacts when screening for isotope pattern and adduct relations using `pattern.search` and `adduct.search`, respectively. For example, consider a satellite peaks having a slightly larger m/z than its monoisotopic parent peaks. Then, a m/z difference from a 13C isotope between monoisotopic parent and M+1 peak often leads to a 15N isotope difference between satellite and M+1 peak. This artifact causes bogus isotope pattern groups (with the satellite peak assigned the monoisotopic peak in this example), group overlaps (see `pattern.search`) and interfering peaks in components (see `combine`).

Still, given the brute approach of `rm.sat`, there is no guarantee that all peaks removed are indeed satellite peaks. As an alternative, one may filter for peaks with an overly short eluation time/scan number or use data from MS devices that are less prone to produce satellite peaks.

Author(s)

Martin Loos

See Also

`peaklist`

Examples

```
data(peaklist);
peaklist<-rm.sat(peaklist,dmz=0.3,drt=0.1,intrat=0.015,spar=0.8,corcut=-1000,plotit=TRUE);
peaklist<-peaklist[peaklist[,4],1:3];
```

Index