Package 'enviPat'

April 8, 2019

Type Package

Title Isotope Pattern, Profile and Centroid Calculation for Mass Spectrometry

Version 2.4

Date 2019-03-18

Author Martin Loos, Christian Gerber

Maintainer Martin Loos <mloos@envibee.ch>

Description Fast and very memory-efficient calculation of isotope patterns, subsequent convolution to theoretical envelopes (profiles) plus valley detection and centroidization or intensoid calculation. Batch processing, resolution interpolation, wrapper, adduct calculations and molecular formula parsing.
Loos, M., Gerber, C., Corona, F., Hollender, J., Singer, H. (2015) <doi:10.1021/acs.analchem.5b00941>.

License GPL-2

BugReports https://github.com/blosloos/enviPat/issues

URL http://www.envipat.eawag.ch/,
http://pubs.acs.org/doi/abs/10.1021/acs.analchem.5b00941

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-04-07 22:12:56 UTC

R topics documented:

`enviPat-package` *Calculation of isotope patterns, stick profiles (envelopes) and centroids/intensoids for mass spectrometry.*

Description

Fast and memory-efficient calculation of isotope patterns (fine structures) for up to very large molecules, based on three different algorithms. Subsequent convolution of isotope patterns with a peak shape function to theoretical envelopes (profiles). Based on envelopes, valley detection and centroidization/intensoid calculation. Allows for batch processing of chemical formulas and interpolation of measurement resolutions. Includes a wrapper combining all of the above functionalities.

Furthermore, includes (1) a check for consistency of chemical formulas, (2) a check for molecules with overlapping isotope patterns, (3) a list of all stable isotopes, (4) a list of different resolution data sets for Thermo Orbitrap and QExactive high-resolution mass spectrometers and (5) a list of adducts formed during electorspray ionization (ESI).

A web-based GUI for enviPat is freely available under `www.envipat.eawag.ch/`.

Details

Package:	enviPat
Type:	Package
Version:	1.0
Date:	2013-03-05
License:	GPL-2
LazyLoad:	yes

Author(s)

Martin Loos, Christian Gerber

Maintainer: Martin Loos <martin.loos@alumni.ethz.ch>

References

Loos, M., Gerber, C., Corona, F., Hollender, J., Singer, H. (2015). Accelerated isotope fine structure calculation using pruned transition trees, Analytical Chemistry 87(11), 5738-5744.

http://pubs.acs.org/doi/abs/10.1021/acs.analchem.5b00941

See Also

check_chemform getR isopattern envelope vdetect isowrap check_several

isotopes resolution_list chemforms

adducts check_ded mergeform subform multiform

---

adducts *Adduct list*

Description

List of common adducts observed for ESI-MS measurements in soft positive and negative ionization modes.

Usage

```
data(adducts)
```

Format

A data frame with 47 observations on the following 6 variables.

`Name` Adduct name

`calc` Equation for calculating adduct m/z from uncharged non-adduct molecular mass M (m/z = M/z + X)

`Charge` z

`Mult` 1/z

`Mass` X

`Ion_mode` Ionization mode (positive or negative)

`Formula_add` Adduct chemical formula to be added

`Formula_ded` Adduct chemical formula to be subtracted

`Multi` Factor to multiply chemical formula with

Details

The correct way to calculate the isotopic pattern of a specific adduct is the following. First, multiply the chemical formula of the molecule by the times it appears in the final adduct; multiform. Second, add the chemical formula of any adduct to that of the molecule; mergeform. Third, subtract the chemical formula of any deduct from that of the molecule; check_ded & subform. Finally, calculate the isotopic fine structure using the correct charge argument in isopattern.

Note

Chemical formulas must conform to what is described in check_chemform.

Source

https://fiehnlab.ucdavis.edu/staff/kind/Metabolomics/MS-Adduct-Calculator/

References

Huang N., Siegel M.M., Kruppa G.H., Laukien F.H., J. Am. Soc. Mass. Spectrom. 1999, 10. Automation of a Fourier transform ion cyclotron resonance mass spectrometer for acquisition, analysis, and e-mailing of high-resolution exact-mass electrospray ionization mass spectral data

See Also

multiform mergeform check_ded subform

Examples

```
# example of M+H adduct batch calculation
data(adducts)
data(isotopes)
data(chemforms)
# (1) check formulas for consistency - recommended
checked_chemforms <-check_chemform(isotopes, chemforms)
# (2) multiply, see column 4 of adducts
chemforms <-multiform(checked_chemforms[,2],1)
# (3) add adduct - see column 7 of adducts
chemforms<-mergeform(chemforms,"H1")
# (4) calculate fine structure
patterns <- isopattern(isotopes, chemforms)
```

---

check_chemform *Check chemical formulas*

Description

Checks chemical formulas (=a vector of character strings) for consistency with usage in isopattern; calculation of the molecular mass.

Usage

```
check_chemform(isotopes,chemforms,get_sorted=FALSE,get_list=FALSE)
```

Arguments

isotopes	isotopes
chemforms	Vector of character strings with chemical formulas
get_sorted	Should elements in each formula be sorted according to their order in isotopes?
get_list	Return list with vectors of elementwise atom counts contained in each chemical formula?

Details

Default checks if (1) a chemical formula contains only letters, numbers and square or round brackets, (2) elements can be found in isotopes and (3) letters and round brackets are all followed by a number of counts. Where (3) are missing, they are set to 1.

(2) must consist of an upper case letter, possibly followed by lower case letters; to refer to individual isotopes (e.g., from isotope labelling of a molecule, e.g., N5 vs. [15]N2N3), square brackets may precede the capital letter. Any other symbols which may be part of a chemical formula (e.g., charges (+), dashes, asterisks, ...) are not permissible.

The molecular mass will be calculated from isotope masses and abundances listed in isotopes.

Value

Dataframe with 3 columns for get_list=FALSE:

warning	Correct chemical formula, FALSE/TRUE?
new_formula	Chemical formula
monoisotopic_mass	Monoisotopic mass

Or list containing vector of elements for get_list=TRUE.

Note

Highly recommended for usage with isopattern

Author(s)

Martin Loos, Christian Gerber

See Also

isopattern isotopes

Examples

```
# Check package data set of chemical formulas #############
data(chemforms);
data(isotopes);
checked<-check_chemform(isotopes,chemforms);
```

```
checked;

# Check for some senseless molecular formulas #############
chemforms<-c("C900Cl4H49","O82394","C8G500Zn9","Br1","6DBr9889");
data(isotopes);
checked<-check_chemform(isotopes,chemforms);
checked;

# Molecular mass with and without isotope labelling #######
chemforms<-c("C10H5N4O5","[13]C2C8D2H3[15]N2N2[18]O2O3");
data(isotopes);
checked<-check_chemform(isotopes,chemforms);
checked;
```

---

check_ded *Check if a chemical formula is subset in another one*

Description

Check if a chemical formula is contained in another chemical formula

Usage

```
check_ded(formulas, deduct)
```

Arguments

formulas	Vector with the containing chemical formula(s)
deduct	Chemical formula to be contained ("deduct")

Value

Returns a vector with length of input `formulas`, with `TRUE` if deduct is not contained and `FALSE` otherwise.

Note

Might be used used prior to subtracting a "deduct" chemical formula from that of a molecule when including adducts in the calculation of isotopic patterns. Chemical formulas must conform to what is described in `check_chemform`.

Author(s)

Martin Loos

See Also

`adducts`

Examples

```
formulas<-c("C8H4Cl2","C10H16O2","C3H10")
deduct<-c("C4H10")
check_ded(formulas, deduct)
```

`check_several` *Check for overlapping molecules.*

Description

Check for molecules overlapping in m/z, based on isotope fine structures from `isopattern` or on centroids/intensoids from `envelope`.

Usage

```
check_several(pattern, dmz, ppm = TRUE)
```

Arguments

`pattern`	Output from `isopattern` or from `envelope`.
`dmz`	m/z window. In combination with `ppm=TRUE` set as ppm or with `ppm=FALSE` set as absolute m/z.
`ppm`	Should m/z window be set in ppm (`TRUE`) or absolute m/z (`FALsE`)?

Details

Overlaps in m/z among molecules are screened for within the m/z tolerance defined by the arguments `dmz` and `ppm`.

Value

Dataframe with 4 columns, with number of rows equal to the length of argument `pattern`

`compound`	Chemical formula of the compound
`warning`	Overlap detected?
`to?`	If overlap: with wich other compound(s)? Refers to row number, recycled for peak\#.
`peak\#`	If overlap: with which peak(s) of the other compound(s)? Refers to peak number.

Author(s)

Martin Loos, Christian Gerber

See Also

isopattern envelope

Examples

```
data(isotopes)
data(chemforms)
pattern<-isopattern(
  isotopes,
  chemforms,
  threshold=0.1,
  plotit=TRUE,
  charge=FALSE,
  emass=0.00054858,
  algo=1
)
check_several(pattern,dmz=0.001,ppm=FALSE)
```

---

chemforms *Set of exemplary chemical formulas for small molecules.*

Description

Vector with character strings of exemplary chemical formulas (pesticides, pharmaceuticals)

Usage

```
data(chemforms)
```

Format

Vector with character strings

Examples

```
data(chemforms)
chemforms
```

envelope *Isotope pattern envelope calculation*

Description

Convolutes an isotope pattern from `isopattern` with a peak shape function (Gaussian or Cauchy-Lorentz function) to its theoretical envelope (profile), at a given measurement resolution. The envelope is represented by sticks, i.e. measurement abundances at discrete m/z intervals.

Usage

```
envelope(pattern, ppm = FALSE, dmz = "get", frac = 1/4, env = "Gaussian",
resolution = 5e+05, plotit = FALSE, verbose = TRUE)
```

Arguments

`pattern`	List of isotope pattern(s) as generated by `isopattern`.
`ppm`	Should stick discretization be set in ppm (`TRUE`) or absolute m/z (`FALSE`)? Only checked if `dmz` is not set to `"get"`; check details section.
`dmz`	Stick discretization. Set to `"get"` to derive discretization from argument `resolution` or set a numerical value in combination with `ppm` to use as ppm or absolute m/z. Check details section.
`frac`	Used if `dmz` is set to `"get"`. Check details section.
`env`	Peak shape function; either `"Gaussian"` or `"CauchyLorentz"`.
`resolution`	Single resolution value or vector of resolutions with length equal to the number of entries in list `pattern`. Check resolution definition in the details section.
`plotit`	Should results be plotted, `TRUE/FALSE` ?
`verbose`	Verbose, `TRUE/FALSE`?

Details

The theoretical profiles are represented by sticks, i.e. abundances at discrete m/z intervals. While the profile width is set by argument `resolution`, the mass discretization between adjacent sticks can be set in two different ways.

On the one hand, discretization can be given as a numerical value, either in ppm or absolute m/z. To do so, set argument `dmz` to a numerical value and specify with argument `ppm` if this value is stating the discretization in ppm or as absolute m/z.

On the other hand, discretization can be derived from the measurement resolution (R) set by argument `resolution`. To do so, set `dmz` to `"get"`, which leads to argument `ppm` being ignored. In this case, the stick discretization is retrieved from (dm/z)*`frac`, with (dm/z) = (m/z)/R = peak width at half maximum.

Value

List with length equal to length of list `pattern`, with equal names of list entries. Each entry in that list contains the sticks of the envelope in two columns:

`m/z`	Stick m/z
`abundance`	Stick abundance

Note

The resolution R is defined as R=(m/z)/(dm/z), with dm/z = peak width at half maximum, cp. `resolution_list`.

Author(s)

Martin Loos, Christian Gerber

References

Li, L., Kresh, J., Karabacak, N., Cobb, J., Agar, J. and Hong, P. (2008). A Hierarchical Algorithm for Calculating the Isotopic Fine Structures of Molecules. Journal of the American Society for Mass Spectrometry, 19, 1867–1874.

See Also

`isopattern` `getR` `vdetect`

Examples

```
############################
# batch of chemforms #######
data(isotopes)
data(chemforms)
chemforms<-chemforms[1:5]

pattern<-isopattern(
  isotopes,
  chemforms,
  threshold=0.1,
  plotit=TRUE,
  charge=FALSE,
  emass=0.00054858,
  algo=2
)

profiles<-envelope(
    pattern,
    ppm=FALSE,
    dmz=0.0001,
    frac=1/4,
    env="Gaussian",
```

```
    resolution=1E6,
    plotit=TRUE
)
############################
```

---

getR *Interpolation of MS measurement resolution*

Description

Given a set of MS measurement resolutions (R) as a function of measurement mass (m/z), getR interpolates R for any given molecular mass(es) calculated by `check_chemform` using `smooth.spline`.

Usage

```
getR(checked, resmass, nknots = 13, spar = 0.1, plotit = TRUE)
```

Arguments

`checked`	Dataframe produced by `check_chemform`.
`resmass`	Dataframe with two columns, resolution and mass; such as the list entries in `resolution_list`.
`nknots`	Integer number of knots to use for the smoothing spline. Default = 6. See also `smooth.spline`.
`spar`	Smoothing parameter, (0,1]. See also `smooth.spline`.
`plotit`	Plot results, `TRUE/FALSE` ?

Value

Vector with resolutions.

Note

`check_chemform` gives molecular masses (m/z) for z=+/-1 only. If z>1 or z<-1 is required, molecular mass entries in argument checked have to be divided accordingly to be consistent.

Author(s)

Martin Loos, Christian Gerber

See Also

`smooth.spline` `check_chemform` `resolution_list`

Examples

```
data(resolution_list)
resmass<-resolution_list[[4]]
data(isotopes)
data(chemforms)
checked<-check_chemform(isotopes,chemforms)
resolution<-getR(checked,resmass,nknots=13,spar=0.1,plotit=TRUE)

# same for z=-2:
checked<-check_chemform(isotopes,chemforms)
checked[,3]<-(checked[,3]/abs(-2))
resolution<-getR(checked,resmass,nknots=13,spar=0.1,plotit=TRUE)
```

`isopattern` *Isotope pattern calculation*

Description

The function calculates the isotopologues ("isotope fine structure") of a given chemical formula or a set of chemical formulas (batch calculation) with fast and memory efficient transition tree algorithms, which can handle relative pruning thresholds. Returns accurate masses, probabilities and isotopic compositions of individual isotopologues. The isotopes of elements can be defined by the user.

Usage

```
isopattern(isotopes, chemforms, threshold = 0.001, charge = FALSE,
emass = 0.00054858, plotit = FALSE, algo=1, rel_to = 0, verbose = TRUE,
return_iso_calc_amount = FALSE)
```

Arguments

`isotopes`	Dataframe listing all relevant isotopes, such as `isotopes`.
`chemforms`	Vector with character strings of chemical formulas, such as data set `chemforms` or the second column in the value of `check_chemform`.
`threshold`	Probability below which isotope peaks can be omitted, as specified by argument `rel_to`. Set to `0` if all peaks shall be calculated.
`charge`	z in m/z. Either a single integer or a vector of integers with length equal to that of argument `chemforms`. Set to `FALSE` for omitting any charge calculations.
`emass`	Electrone mass; only relevant if `charge` is not set to `FALSE`.
`plotit`	Should results be plotted, `TRUE/FALSE`?
`algo`	Which algorithm to use? Type `1` or `2`. See details.
`rel_to`	Probability definition, numeric `0,1,2,3 or 4`? See details.
`verbose`	Verbose, `TRUE/FALSE`?
`return_iso_calc_amount`	Ignore; number of intermediate isotopologues.

Details

Isotope pattern calculation can be done by chosing one of two algorithms, set by argument `algo`. Both algorithms use transition tree updates to derive the exact mass and probability of a new isotopologue from existing ones, by steps of single isotope replacements. These transition tree approaches are memory-efficient and fast for a wide range of molecular formulas and are able to reproduce the isotope fine structure of molecules. The latter must often be pruned during calculation, c.p. argument `rel_to`.

`algo=1` grows transition trees within element-wise sub-molecules, whereas `algo==2` grows them in larger sub-molecules of two elements, if available. The latter approach can be slightly more efficient for very large or very complex molecules. The sub-isotopologues within sub-molecules are finally combined to the isotopologuees of the full molecule. In contrast, intermediate counts of sub-isotopologues instead of fine structures are returned for `return_iso_calc_amount==TRUE`

`rel_to` offers 5 possibilities of how probabilities are defined and pruned, each affecting the `threshold` argument differently. Default option `rel_to=0` prunes and returns probabilities relative to the most intense isotope peak; `threshold` states a percentage of the intensity of this latter peak. Similarly, option `rel_to=1` normalizes relative to the peak consisting of the most abundant isotopes for each element, which is often the monoisotopic one. Option `rel_to=2` prunes and returns absolute probabilities ; `threshold` is not a percentage but an abolute cutoff. Options `rel_to=3` and `rel_to=4` prune relative to the most intense and "monoisotopic" peak, respectively. Although `threshold` is a percentage, both options return absolut probabilities .

Value

List with length equal to length of vector `chemforms`; names of entries in list = chemical formula in chemform. Each entry in that list contains information on individual isotopologues (rows) with columns:

`m/z`	First column; m/z of an isotope peak.
`abundance`	Second column; abundance of an isotope peak. Probabilities are set relative to the most abundant peak of the isotope pattern.
`12C, 13C, 1H, 2H, ...`	Third to all other columns; atom counts of individual isotopes for an isotope peak.

warning

Too low values for `threshold` may lead to unnecessary calculation of low probable isotope peaks - to the extent that not enough memory is available for either of the two algorithms.

Note

It is highly recommended to check argument `chemforms` with `check_chemform` prior to running `isopattern`; argument `chemforms` must conform to chemical formulas as defined in `check_chemform`. Element names must be followed by numbers (atom counts of that element), i.e. C1H4 is a valid argument whereas CH4 is not. Otherwise, numbers may only be used in square brackets to denote individual isotopes defined in the element name column of iso_list, such as [14]C or [18]O. For example, [13]C2C35H67N1O13 is the molecular formula of erythromycin labeled at two C-positions with [13]C; C37H67N1O13 is the molecular formula of the unlabeled compound.

For correct adduct isotope pattern calculations, please check adducts.

Author(s)

Martin Loos, Christian Gerber

References

Loos, M., Gerber, C., Corona, F., Hollender, J., Singer, H. (2015). Accelerated isotope fine structure calculation using pruned transition trees, Analytical Chemistry 87(11), 5738-5744.

http://pubs.acs.org/doi/abs/10.1021/acs.analchem.5b00941

http://www.envipat.eawag.ch/index.php

See Also

isopattern chemforms check_chemform getR envelope vdetect check_several

Examples

```
############################
# batch of chemforms #######
data(isotopes)
data(chemforms)
pattern<-isopattern(
  isotopes,
  chemforms,
  threshold=0.1,
  plotit=TRUE,
  charge=FALSE,
  emass=0.00054858,
  algo=1
)
############################
# Single chemical formula ##
data(isotopes)
pattern<-isopattern(
  isotopes,
  "C100H200S2Cl5",
  threshold=0.1,
  plotit=TRUE,
  charge=FALSE,
  emass=0.00054858,
  algo=1
)
############################
```

isotopes *Stable isotopes*

Description

Dataframe with stable isotopes.

Usage

```
data(isotopes)
```

Format

A data frame with 302 observations on the following 4 variables.

`element` Chemical element

`isotope` Stable isotopes of an element

`mass` Relative atomic mass

`abundance` Isotopic composition of an element

`ratioC` Maximum number of atoms of an element for one C-atom in a molecule, based on 99.99 % of case molecules.

Details

The `ratioC`-value stems from a database survey conducted by Kind&Fiehn (2007); to disable, set value to 0. The list serves as input into several package nontarget-functions. The first column of the data frame also contains names of specific isotopes used for labeled compounds.

Source

http://physics.nist.gov/cgi-bin/Compositions/stand_alone.pl

References

Kind, T. and Fiehn, O., 2007. Seven golden rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. BMC Bioinformatics, 8:105.

Examples

```
data(isotopes)
```

isowrap *Combined (batch) calculation of isotope pattern, envelope and centroids/intensoids/valleys on interpolated resolutions.*

Description

Wrapper combining the functions getR, isopattern, envelope and vdetect.

Uses chemical formulas from check_chemform as argument.

Usage

```
isowrap(isotopes, checked, resmass, resolution = FALSE, nknots = 6,
spar = 0.2, threshold = 0.1, charge = 1, emass = 0.00054858, algo=2,
ppm = FALSE, dmz = "get", frac = 1/4, env = "Gaussian",
detect = "centroid", plotit = FALSE)
```

Arguments

isotopes	Dataframe listing all relevant isotopes, such as isotopes.
checked	Output dataframe from check_chemform with correct chemical formulas.
resmass	For resolution interpolation: dataframe with two columns, resolution and mass; see getR. Otherwise, set to FALSE and use argument resolution to utilize a single resolution.
resolution	Single resolution value. Only used if argument resmass is set to FALSE.
nknots	Number of knots, see getR. Ignored if argument resmass set to FALSE.
spar	Smoothing parameter, see getR. Ignored if argument resmass set to FALSE.
threshold	Abundance below which isotope peaks are omitted, see isopattern.
charge	z in m/z, see isopattern.
emass	Electrone mass. Only relevant if charge is not set to FALSE, see isopattern.
algo	Which algorithm to use? Type 1 or 2. See details section in isopattern.
ppm	Set stick discretization, see details section of envelope.
dmz	Set stick discretization, see details section of envelope.
frac	Set stick discretization, see details section of envelope.
env	Peak shape function, see envelope.
detect	Return either "centroid", "intensoid" or "valley". See vdetect.
plotit	Should results be plotted, TRUE/FALSE?

Value

List with length equal to length of list profiles, with equal names of list entries. Each entry in that list contains the centroids, intensoids or valley of the envelope in two columns:

m/z	m/z
abundance	area(centroid) or abundance (intensoid, valley)

Author(s)

Martin Loos, Christian Gerber

See Also

vdetect

Examples

```
data(isotopes);
data(resolution_list);
data(chemforms);
chemforms<-chemforms[1:10];

checked<-check_chemform(
    isotopes,
    chemforms
  );

resmass<-resolution_list[[1]]

centro<-isowrap(
  isotopes,
  checked,
  resmass=resolution_list[[4]],
  resolution=FALSE,
  nknots=4,
  spar=0.2,
  threshold=0.1,
  charge=1,
  emass=0.00054858,
  algo=2,
  ppm=FALSE,
  dmz="get",   # retrieve dm from R=m/dm
  frac=1/4,
  env="Gaussian",
  detect="centroid",
  plotit=TRUE
)
```

---

mergeform *Combine chemical formulas*

Description

Combine chemical formulas

Usage

```
mergeform(formula1,formula2)
```

Arguments

formula1	Vector of first chemical formula(s), character string(s)
formula2	Second chemical formula, single character string

Details

Useful for adduct calculations, check adducts. Chemical formulas must conform to what is described in check_chemform.

Value

Merged chemical formula(s), character string

Author(s)

Martin Loos

See Also

adducts

Examples

```
formula1<-c("C10[13]C2H10Cl10")
formula2<-c("C2H5Na1")
mergeform(formula1,formula2)
```

---

multiform *Multiply a chemical formula*

Description

Multiply all atom numbers in a chemical formula by a factor

Usage

```
multiform(formula1,fact)
```

Arguments

formula1	Chemical formula to be multiplied, vector of character strings
fact	Factor to multiply with

Details

Useful for adduct calculations, check adducts. Chemical formulas must conform to what is described in check_chemform.

Value

Multiplied chemical formula, character string

Author(s)

Martin Loos

See Also

adducts

Examples

```
formula1<-c("C10[13]C2H10Cl10")
multiform(formula1,3)
```

---

resolution_list *Resolutions (R) list for mass spectrometers*

Description

List of different resolutions R=f(m/z) for various (high-resolution) mass spectrometers. For each of the instruments, different resolution settings are available. Here, R is defined as R=(m/z)/(dm/z), with dm/z = peak width at half maximum. Serves as input to getR to interpolate R from given molecular masses.

Usage

```
data(resolution_list)
```

Format

The format is: List with 29 data sets: Instrument_(massRange_instrumentMode_slicerMode)_Resolution@m/z

Elite/R240000@400

Elite/R120000@400

Elite/R60000@400

Elite/R30000@400

OrbitrapXL,Velos,VelosPro/R120000@400

OrbitrapXL,Velos,VelosPro/R60000@400

OrbitrapXL,Velos,VelosPro/R30000@400

OrbitrapXL,Velos,VelosPro/R15000@400

OrbitrapXL,Velos,VelosPro/R7500@400

Q-Exactive,ExactivePlus/280K@200

Q-Exactive,ExactivePlus/R140000@200

Q-Exactive,ExactivePlus/R70000@200

Q-Exactive,ExactivePlus/R35000@200

Q-Exactive,ExactivePlus/R17500@200

Exactive/R100000@200

Exactive/R50000@200

Exactive/R25000@200

Exactive/R12500@200

OTFusion,QExactiveHF/480000@200

OTFusion,QExactiveHF/240000@200

OTFusion,QExactiveHF/120000@200

OTFusion,QExactiveHF/60000@200

OTFusion,QExactiveHF/30000@200

OTFusion,QExactiveHF/15000@200

QTOF_XevoG2-S/R25000@200

Sciex_TripleTOF5600_R25000@200

Sciex_TripleTOF6600_R25000@200

Sciex_QTOFX500R_R25000@200

Agilent_low_extended_highSens_QTOF6550_R25000@200

Source

Data assembled from individual measurements.

Examples

```
data(resolution_list)
resolution_list
```

subform *Subtract one chemical formula from another*

Description

Subtract one chemical formula from another

Usage

```
subform(formula1,formula2)
```

Arguments

formula1	Chemical formula to subtract from
formula2	Chemical formula to subtract

Details

Useful for adduct calculations, check adducts. Chemical formulas must conform to what is described in check_chemform. Prior check if formula2 is contained in formula2 at all? See check_ded.

Value

A unified and filtered peaklist

Author(s)

Martin Loos

See Also

adducts,check_ded

Examples

```
formula1<-c("C10[13]C2H10Cl10")
formula2<-c("C2H5[13]C1")
subform(formula1,formula2)
```

vdetect *Valley detection and centroidization*

Description

Checks envelopes calculated by envelope for valleys and extracts centroids or intensoids.

Usage

```
vdetect(profiles,detect="centroid",plotit=TRUE,verbose=TRUE)
```

Arguments

profiles	List of stick profiles as generated by envelope.
detect	To return either "centroid", "intensoid" or "valley".
plotit	Should results be plotted, TRUE/FALSE?
verbose	Verbose, TRUE/FALSE?

Value

List with length equal to length of list profiles, with equal names of list entries. Each entry in that list contains the centroids, intensoids or valleys of the envelope in two columns:

m/z	m/z
abundance	Area (centroid) or abundance (intensoid, valley)

definitions

Valley: local profile minimum, i.e. any envelope stick flanked by two other sticks of higher abundance.

Stick: see envelope.

Centroid mass: intensity-weighted sum of the m/z of sticks between two valleys.

Centroid intensity: profile area between two valleys (mean of upper and lower sum of stick intensities), normalized to the maximum centroid area of the envelope.

Intensoid mass: m/z of the most intense stick between two valleys.

Intensoid intensity: intensity of the most intensive stick between two valleys, normalized to the most intense intensoid.

Note

Too low stick discretization leads to imprecision in valley, centroid and intensoid characteristics.

Author(s)

Martin Loos, Christian Gerber

See Also

isopattern envelope

Examples

```
############################
# batch of chemforms #######
data(isotopes)
data(chemforms)
chemforms<-chemforms[1:5]

pattern<-isopattern(
  isotopes,
  chemforms,
  threshold=0.1,
  plotit=TRUE,
  charge=FALSE,
  emass=0.00054858,
  algo=2
)

profiles<-envelope(
    pattern,
    ppm=FALSE,
    dmz=0.0001,
    frac=1/4,
    env="Gaussian",
    resolution=1E6,
    plotit=TRUE
)

centro<-vdetect(
  profiles,
  detect="centroid",
  plotit=TRUE
)

############################
```

Index