Package ‘InterpretMSSpectrum’

May 3, 2018

Type Package
Title Interpreting High Resolution Mass Spectra
Version 1.2
Date 2018-05-03
Author Jan Lisec [aut, cre]
Maintainer Jan Lisec <jan.lisec@bam.de>
Description Annotate and interpret deconvoluted mass spectra (mass*intensity pairs) from high resolution mass spectrometry devices.
URL http://dx.doi.org/10.1021/acs.analchem.6b02743
License GPL-3
Depends R(>= 2.10.0)
Imports Rdisop, enviPat, plyr, utils, foreach, doParallel, DBI, graphics, grDevices, RSQLite
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-05-03 12:03:25 UTC

R topics documented:

CountChemicalElements .................................................... 2
findMAIN .............................................................................. 2
GenerateMetaboliteSQLiteDB ................................................. 4
GetGroupFactor ............................................................. 5
IMS_parallel .................................................................. 5
InterpretMSSpectrum ...................................................... 6
mScore ................................................................. 8
neutral_losses .............................................................. 9
PlotSpec ........................................................................ 10
sendToMSF ..................................................................... 12
test_spectrum .............................................................. 13
CountChemicalElements

**Description**

CountChemicalElements will split a character (chemical formula) into its elements and count their occurrence.

**Usage**

CountChemicalElements(x = NULL, ele = NULL)

**Arguments**

- **x** Chemical formula.
- **ele** Character vector of elements to count particularly or counting all contained if `NULL`.

**Details**

No testing for any chemical alphabet is performed. Elements may occur several times and will be summed up in this case without a warning.

**Value**

A named numeric with counts for all contained or specified elements.

findMAIN

**Description**

findMAIN will evaluate an ESI spectrum for the potential main adducts, rank obtained suggestions and allow the deduction of the neutral mass of the measured molecule.

**Usage**

findMAIN(spec, adductmz = NULL, ionmode = c("positive", "negative")[1],
          adducthyp = NULL, ms2spec = NULL, rules = NULL, mzabs = 0.01,
          ppm = 5, mainpkthr = 0.005, collapseResults = TRUE)
findMAIN

Arguments

spec A mass spectrum. Either a matrix or data frame, the first two columns of which are assumed to contain the 'mz' and 'intensity' values, respectively.

adductmz Manually specified peak for which adducthyp should be tested, or 'NULL' (default), to test all main peaks. What is a main peak, is governed by mainpkthr.

ionmode Ionization mode, either "positive" or "negative". Can be abbreviated.

adducthyp Adduct hypotheses to test for each main peak. Defaults to c("[M+H]+", "[M+Na]+", "[M+K]+") for positive mode and c("[M-H]-", "[M+CI]-", "[M+HCOOH-H]-")..

ms2spec Second spectrum limiting main peak selection. If available, MS^nE or bbCID spectra may allow further exclusion of false positive adduct ions, as ions of the intact molecule (protonated molecule, adduct ions) should have lower intensity in the high-energy trace than in low-energy trace.

rules Adduct/fragment relationships to test, e.g. c("[M+Na]+", "[M+H-H2O]+"), or 'NULL' for default set (see Adducts)

mzabs Allowed mass error, absolute (Da).

ppm Allowed mass error, relative (ppm), which is added to 'mzabs'.

mainpkthr Intensity threshold for main peak selection, relative to base peak.

collapseresults If a neutral mass hypothesis was found more than once (due to multiple adducts suggesting the same neutral mass), return only the one with the highest adduct peak. Should normally kept at TRUE, the default.

Details

Electrospray ionization (ESI) mass spectra frequently contain a number of different adduct ions, multimers and in-source fragments ([M+H]+, [M+Na]+, [2M+H]+, [M+H-H2O]+), making it difficult to decide on the compound’s neutral mass. This function aims at determining the main adduct ion and its type (protonated, sodiated etc.) of a spectrum, allowing subsequent database searches e.g. using MS-FINDER, SIRIUS or similar.

Value

A list-like 'findMAIN' object for which 'print', 'summary' and 'plot' methods are available.

References


Examples

utils::data(esi_spectrum, package = "InterpretMSSpectrum")
fmr <- findMAIN(esi_spectrum)
plot(fmr)
head(summary(fmr))
InterpretMSSpectrum(fmr[[1]], precursor=263, ionization="ESI+", formula_db="ESI.db")
GenerateMetaboliteSQLiteDB

GenerateMetaboliteSQLiteDB will set up a SQLite data base containing potential metabolite formulas, their masses and isotopic distribution for use with InterpretMSSpectrum.

Usage

GenerateMetaboliteSQLiteDB(dbfile = "SQLite_APCI.db", ionization = c("APCI", "ESI"), mass_range = c(80, 160), ncores = 2)

Arguments

dbfile path and filename of the final SQLiteDB.
ionization Has to be specified to account for different plausibility rules and elemental composition.
mass_range For testing use default range, otherwise use your measurement range.
ncores Number of cores. Use as many as possible.

Details

The process takes a long time for larger masses (>400 Da). Parallel processing with 8 cores is highly recommended. Alternatively pre-processed versions can be downloaded on request to <jan.lisec@charite.de>. To process a 1 Da range (from 900 to 901) for ESI does take ~5 minutes on 8 cores.

Value

Returns NULL invisible. Will write an SQL_DB as specified by 'dbfile'.

Examples

#this would be relatively fast, but for higher masses it is getting much slower

GenerateMetaboliteSQLiteDB(dbfile="APCI.db", ionization="APCI", mass_range=c(80,140), ncores=1)
GenerateMetaboliteSQLiteDB(dbfile="ESI.db", ionization="ESI", mass_range=c(900,901), ncores=8)
GetGroupFactor

Description

GetGroupFactor will split a numeric vector according to a specified gap value. This is often a useful tool and therefore exported to the namespace.

Usage

GetGroupFactor(x, gap)

Arguments

- x: Numeric vector.
- gap: Difference between two consecutive values at which a split is generated.

Value

A factor vector of length(x) indicating the different groups in x.

Examples

```r
x <- c(1:3, 14:12, 6:9)
GetGroupFactor(x=x, gap=2)
split(x, GetGroupFactor(x=x, gap=2))
```

IMS_parallel

Description

IMS_parallel is a parallel implementation of InterpretMSSpectrum.

Usage

IMS_parallel(spectra = NULL, ncores = 8, precursor = NULL,
 correct_peak = NULL, ...)

Arguments

- spectra: List of spectra.
- ncores: Number of cores available.
- precursor: vector of precursor masses of length(spectra).
- correct_peak: Potentially a vector of correct Peaks, see InterpretMSSpectrum for details.
- ...: Further parameters passed directly to InterpretMSSpectrum.
InterpretMSSpectrum

Details

For mass processing and testing it may be sufficient to use InterpretMSSpectrum without plotting functionality. However, function is likely to be deprecated or integrated as an option into the main function in the future.

Value

A list of InterpretMSSpectrum result objects which can be systematically evaluated. However, note that plotting is unfortunately not enabled for parallel processing.

See Also

InterpretMSSpectrum

InterpretMSSpectrum

Interpreting High-Res-MS spectra.

Description

InterpretMSSpectrum will read, evaluate and plot a deconvoluted mass spectrum (mass*intensity pairs) from either TMS-derivatized GC-APCI-MS data or ESI+/− data. The main purpose is to identify the causal metabolite or more precisely the sum formula of the molecular peak by annotating and interpreting all visible fragments and isotopes.

Usage

InterpretMSSpectrum(spec = NULL, precursor = NULL, correct_peak = NULL, met_db = NULL, typical_losses_definition = NULL, silent = FALSE, dppm = 3, score_cutoff = 0.5, neutral_loss_cutoff = NULL, ionization = c("APCI", "ESI+", "ESI-"))[1], quick_isos = TRUE, formula_db = NULL)

Arguments

spec A 2-column matrix of mz/int pairs. If spec=NULL then InterpretMSSpectrum tries to read data from clipboard (i.e. two columns copied from an Excel spreadsheet).

precursor The ion (m/z) from spec closest to this mass will be considered as precursor (can be nominal, i.e. if precursor=364 then 364.1234 would be selected from spectrum if it is closest).

correct_peak For testing purposes. A character in the form of "name, formula, mz" to evaluate spectra against. Note! Sperating character is ','.

met_db A metabolite DB (e.g. GMD or internal) can be provided to search for candidates comparing M+H ions (cf. Examples).
typical_losses_definition

A file name (e.g. D:/BuildingBlocks_GCAPCI.txt) from where to load relevant neutral losses (cf. Details). Alternatively a dataframe with columns Name, Formula and Mass.

silent

Logical. If TRUE no plot is generated and no output except final candidate list is returned.

dppm

Specifies ppm error for Rdisop formula calculation.

score_cutoff

Specifies initial filtering step threshold per fragment. Sum Formulas with score_i < score_cutoff*max(score) will be removed.

neutral_loss_cutoff

Specifies the allowed deviation in mDa for neutral losses to be accepted from the provided neutral loss list. If NULL determined dependent on ionization.

ionization

Currently 'APCI' or 'ESI-' or 'ESI+' are supported as ionization modes.

quick_isos

There are two ways to compute isotope pattern for comparison, a fast but error prone using Rdisop and a slow but (more) correct using enviPat.

formula_db

A predetermined database of sum formulas and their isotopic fine structures can be used to extremely speed up the function.

Details

For further details refer to and if using please cite Jaeger et al. (http://dx.doi.org/10.1021/acs.analchem.6b02743) in case of GC-APCI and Jaeger et al. 2017 (RCM, accepted) for ESI data. The Interpretation is extremely speed up if 'formula_db' (a predetermined database of potential sum formulas) is provided within the function call. Within the package you may use GenerateMetaboLiteSQLiteDB to prepare one for yourself or request a download link from jan.lisec@charite.de as de novo calculation may take several days.

Value

An annotated plot of the mass spectrum and detailed information within the console. Main result, list of final candidate formulas and their putative fragments, will be returned invisibly.

Examples

```r
# load test data
tests::data(apci_spectrum)

# provide information of a correct peak (if you know)
correct_peak <- c("Glutamic acid (3TMS), C14H33NO4Si3, 364.1790")

# provide database of known peaks and correct peak
met_db <- data.frame("Name"=c("Glutamic acid (3TMS)"), "other peak with same sum formula"),
"Formula"=c("C14H33NO4Si3", "C14H33NO4Si3"),
"M+H"=c(364.179,364.179), stringsAsFactors=FALSE, check.names=FALSE)

# apply function providing above arguments (dppm is set to 0.5 to reduce run time)
res <- InterpretMSSpectrum(spec=apci_spectrum, correct_peak=correct_peak, met_db=met_db, dppm=0.5)

# show final function result (score-sorted list of potential fragment trees)
```
mScore

Description

mScore will calculate a mass defect weighted score for an mz/int values measure for an isotopic cluster in comparison to the theoretically expected pattern.

Usage

mScore(obs = NULL, the = NULL, dabs = 5e-04, dppm = 2, int_prec = 0.02, limit = 0, rnd_prec = 0)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs</td>
<td>Observed (measured) values, a matrix with two rows (mz/int).</td>
</tr>
<tr>
<td>the</td>
<td>Theoretical (estimated from sum formula) values, a matrix with two rows (mz/int).</td>
</tr>
<tr>
<td>dabs</td>
<td>Absolute allowed mass deviation (the expected mass precision will influence mScore – see Details).</td>
</tr>
<tr>
<td>dppm</td>
<td>Relative allowed mass deviation (the expected mass precision will influence mScore – see Details).</td>
</tr>
<tr>
<td>int_prec</td>
<td>The expected intensity precision will influence mScore (see Details).</td>
</tr>
<tr>
<td>limit</td>
<td>minimal value of mScore. Should be left on zero.</td>
</tr>
<tr>
<td>rnd_prec</td>
<td>Rounding precision of mScore.</td>
</tr>
</tbody>
</table>

Details

The maximum expected average mass error should be specified in ppm. A observed pattern deviating that much from the theoretical pattern would still receive a reasonable (average) mScore while observations deviating stronger or less strong will reach lower or higher mScores respectively. Likewise the intensity precision should specify the average quality of your device to maintain stable isotopic ratios.
neutral_losses

Value

Scalar mScore giving the quality of the observed data if theoretical data are true.

Examples

```r
# get theoretical isotopic pattern of Glucose
glc <- Rdop::getMolecule("C6H12O6")$isotopes[[1]][1:3]
mScore(obs=glc, the=glc)
# modify pattern by maximum allowable error (2ppm mass error, 2% int error)
glc_theoretic <- glc
glc[1,] <- glc[1,]+2*glc[1,]/10^6
glc[2,1:2] <- c(-0.02,0.02)*glc[2,1:2]
mScore(obs=glc, the=glc_theoretic)

# simulate mass and int defects
ef <- function(x, e) {runif(1,x-x*e,x*x*e)}
glc_obs <- glc
glc_obs[1,] <- sapply(glc[1,], ef, e=2*10^-6)
glc_obs[2,] <- sapply(glc[2,], ef, e=0.02)
mScore(obs=glc_obs, the=glc)

# simulate mass and int defects systematically
ef <- function(x, e) {runif(1,x-x*e,x*x*e)}
n <- 11
mz_err <- round(seq(0,5,length.out=n),3)
int_err <- round(seq(0,0.1,length.out=n),3)
mat <- matrix(NA, ncol=n, nrow=n, dimnames=list(mz_err, 100*int_err))
glc_obs <- glc
for (i in 1:n) {
  glc_obs[1,] <- sapply(glc[1,], ef, e=mz_err[i]*10^-6)
  for (j in 1:n) {
    glc_obs[2,] <- sapply(glc[2,], ef, e=int_err[j])
    mat[i,j] <- mScore(obs=glc_obs, the=glc)
  }
}
plot(x=1:n, y=1:n, type="n", axes=FALSE, xlab="mass error [ppm]", ylab="isoratio error [%]")
axis(3,at=1:n,rownames(mat),las=2); axis(4,at=1:n,colnames(mat),las=2); box()
cols <- grDevices::colorRampPalette(colors=c(2,6,3))(diff(range(mat))+1)
cols <- cols[mat-min(mat)+1]
text(x=rep(1:n,each=n), y=rep(1:n, times=n), labels=as.vector(mat), col=cols)
```

neutral_losses

Description

A data table defining neutral losses in GC-APCI-MS for silylated compounds.
Usage

data("neutral_losses")

Format

A data frame with 22 observations on the following 3 variables.

Name a character vector
Formula a character vector
Mass a numeric vector

Details

The data frame consists 2 character columns ("Name" and "Formula") and the numeric column "Mass". In a mass spectrum peak pairs are analyzed for mass differences similar to the ones defined in neutral_losses. If such a mass difference is observed we can assume that the according "Formula" is the true neutral loss observed in this spectrum. In a plot this peak pair would be connected by a grey line and annotated with the information from "Name".

Source

This list has been put together manually by Jan Lisec analyzing several GC-APCI-MS data sets.

References

See published paper in $$

Examples

data(neutral_losses)
str(neutral_losses)

PlotSpec

Plot Mass Spectrum.

Description

PlotSpec will read, evaluate and plot a deconvoluted mass spectrum (mass*intensity pairs) from TMS-derivatized GC-APCI-MS data. The main purpose is to visualize the relation between deconvoluted masses.

Usage

PlotSpec(x = NULL, masslab = 0.1, rellab = FALSE, cutoff = 0.01,
cols = NULL, txt = NULL, mz_prec = 4, ionization = NULL,
neutral_losses = NULL, neutral_loss_cutoff = NULL, substitutions = NULL,
xlim = NULL, ylim = NULL)
**Arguments**

- **x**: A two-column matrix with ("mz", "int") information.
- **masslab**: The cutoff value (relative to basepeak) for text annotation of peaks.
- **rellab**: TRUE/FALSE. Label masses relative to largest mass in plot (if TRUE), absolute (if FALSE) or to specified mass (if numeric).
- **cutoff**: Show only peaks with intensity higher than cutoff*I(base peak). This will limit the x-axis accordingly.
- **cols**: Color vector for peaks with length(cols)==nrow(x).
- **txt**: Label peaks with specified text (column 1 specifies x-axis value, column 2 specifies label).
- **mz_prec**: Numeric precision of m/z (=number of digits to plot).
- **ionization**: Either APCI or ESI (important for main peak determination).
- **neutral_losses**: Data frame of defined building blocks (Name, Formula, Mass). If not provided data("neutral_losses") will be used.
- **neutral_loss_cutoff**: Specifies the allowed deviation in mDa for neutral losses to be accepted from the provided neutral loss list.
- **substitutions**: May provide a two column table of potential substitutions (for adducts in ESI-MS).
- **xlim**: To specify xlim explicitly (for comparative plotting).
- **ylim**: To specify ylim explicitly (for comparative plotting).

**Value**

An annotated plot of the mass spectrum.

**Examples**

```r
#load test data and apply function
utils::data(apci_spectrum, package = "InterpretMSSpectrum")
PlotSpec(x=apci_spectrum, ionization="APCI")

# normalize test data by intensity
s <- apci_spectrum
s[,2] <- s[,2]/max(s[,2])
PlotSpec(x=s)

# use relative labelling
PlotSpec(x=s, rellab=364.1789)

# avoid annotation of masses and fragments
PlotSpec(x=s, masslab=NULL, neutral_losses=NA)

# provide individual neutral loss set
tmp <- data.frame("Name"=c("Loss1","Loss2"),"Formula"=c("",""),"Mass"=c(90.05,27.995))
PlotSpec(x=s, neutral_losses=tmp)
```
# provide additional color and annotation information per peak
PlotSpec(x=s, cols=1+(s[,2]>0.1), txt=data.frame("x"=s[,2]>0.1,"txt"="txt"))

# simulate a Sodium adduct to the spectrum (and annotate using substitutions)
p <- which.max(s[,2])
s <- rbind(s, c(21.98194+s[p,1], 0.6*s[p,2]))
PlotSpec(x=s, substitutions=matrix(c("H1","Na1"),ncol=2,byrow=TRUE))

# load ESI test data and apply function
utils::data(esi_spectrum)
PlotSpec(x=esi_spectrum, ionization="ESI")

---

## sendToMSF

Exporting spectra to MSFinder.

### Description

Send spectrum to MSFinder.

### Usage

```r
sendToMSF(x, ...)
```

```
## Default S3 method:
sendToMSF(x, precursormz, precursortype = "[M+H]+", 
  outfile = NULL, MSFexe = NULL, ...)

## S3 method for class 'findMAIN'
sendToMSF(x, rank = 1, ms2spec = NULL, outfile = NULL, 
  MSFexe = NULL, ...)
```

### Arguments

- **x**: A matrix or ‘findMAIN’ object
- **...**: Arguments passed to methods of `writeMSF`
- **precursormz**: m/z of (de)protonated molecule or adduct ion
- **precursortype**: adduct type, e.g. "[M+H]+", "[M+Na]+". Accepted values are all adduct ions supported by MSFINDER.
- **outfile**: Name of MAT file. If NULL, a temporary file is created in the per-session temporary directory (see `tempdir`).
- **MSFexe**: Full path of MS-FINDER executable. This needs to be set according to your system. If NULL, MAT files are written but the program is not opened.
- **rank**: Which rank from ‘findMAIN’ should be exported.
- **ms2spec**: An (optional) MS2 spectrum to be passed to MSFINDER. If NULL, the MS1 spectrum used by ‘findMAIN’ is used. If dedicated MS2 spectra are available, this option should be used.
Details
In the default case `x` can be a matrix or data frame, where the first two columns are assumed to contain the 'mz' and 'intensity' values, respectively. Further arguments 'precursorMz' and 'precursorSortype' are required in this case. Otherwise 'x' can be of class findMAIN.

Value
Full path of generated MAT file (invisibly).

References

Examples

```r
utils::data(esi_spectrum, package = "InterpretMSSpectrum")
fmr <- findMAIN(esi_spectrum)
getwd()
sendToMSF(fmr, outfile="tmp.mat")
sendToMSF(fmr, outfile="tmp.mat", rank=1:3)
```

---

description
Example spectrum of Glutamic acid (3TMS) measured on a Bruker impact II.

Usage
data("test_spectrum")

Format
A data frame with 47 observations on the following 2 variables.

- **mz** a numeric vector
- **int** a numeric vector

Examples
data(test_spectrum)
head(test_spectrum)
PlotSpec(test_spectrum)
Index

*Topic datasets
  neutral_losses, 9
  test_spectrum, 13

Adducts, 3

CountChemicalElements, 2

findMAIN, 2

GenerateMetaboliteSQLiteDB, 4, 7
GetGroupFactor, 5

IMS_parallel, 5
InterpretMSSpectrum, 4–6, 6

mScore, 8

neutral_losses, 9

PlotSpec, 10

sendToMSF, 12

tempdir, 12
test_spectrum, 13

writeMSF, 12