Package 'enviPick'

June 6, 2016

Type Package

Title Peak Picking for High Resolution Mass Spectrometry Data

Version 1.5

Date 2016-06-03

Author Martin Loos

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

Description Sequential partitioning, clustering and peak detection of centroided LC-MS mass spectrometry data (.mzXML). Interactive result and raw data plot.

License GPL (>= 2)

Depends R (>= 3.0.1), shiny(>= 0.7.0), readMzXmlData(>= 2.7)

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-06-06 17:56:38

R topics documented:

enviPick-package *Peak picking for high resolution liquid chromatography - mass spectrometry (LC-HRMS) data*

Description

Peak picking for centroided and baseline-corrected high-resolution LC-MS data.mzXML. Built on a three-step approach of (1) data partitioning, (2) unsupervised clustering of extracted ion chromatograms (EIC) and (3) shape-independent peak detection wihtin individual EICs. Interactive plot access to all results and the underlying raw measurements. Browser UI for non-R users. Batch processing.

Details

Package:	enviPick
Type:	Package
Version:	1.0
Date:	2014-14-07
License:	GPL-2

After initial upload of an .mzXML file with `readMSdata`, above steps (1) to (3) are calculated by `mzagglom`, `mzclust` and `mzpick`, respectively. The wrapper for joint upload and processing is `enviPickwrap`. The raw data and the results of each step, a so-called MSlist object, can be viewed by `plotMSlist`, producing an interactive plot that conveniently offers zoom, drag and select functionality and easy identification of individual partitions, EIC cluster or peaks. Batch processing can be done via `enviPickbatch`. To export a peak list from an MSlist object, use `writePeaklist`.

For converting LC-HRMS measurement files from various vendor formats or .mzML to centroided .mzXML we strongly recommend the MSConvert tool from ProteoWizard; for centroidization choose Filters -> Peak Picking -> Prefer Vendor -> Add.

Warning

This package has only been tested on HIGH-RESOLUTION Thermo Orbitrap and QExactive measurements processed (centroided) with ProteoWizard‘s MSConvert. It may not give satisfying results for chromatograms affected by mass shifts from centroid-centroid interferences prevalent at low resolutions.

Note

In the package context, peak picking refers to extracting individual ion chromatograms (EICs) from centroided data and identifying peaks in these EICs. In the ProteoWizard MSConvert context, peak picking refers to identifying individual peaks within single HRMS scans, alias centroidization.

Author(s)

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

References

Loos, M. (XXXX). Extraction of ion chromatograms by unsupervised clustering of high-resolution mass spectrometry data. Some Journal. Sometime.

ProteoWizard: Open Source Software for Rapid Proteomics Tools Development Darren Kessner; Matt Chambers; Robert Burke; David Agus; Parag Mallick Bioinformatics 2008 `http://proteowizard.sourceforge.net/`

See Also

`readMSdata` `mzagglom` `mzpart` `mzclust` `mzpick` `plotMSlist` `enviPickwrap` `enviPickbatch`

Examples

```
## Not run:

######################################################
# (1) Define path to an LC-HRMS .mzML file (not provided with package):

filepath.mzML<-"C:/.../2012_07_01.mzML"

# (2) Initialize an MSlist object and load this .mzML file into it:

MSlist<-readMSdata(filepath.mzML, MSlevel=c(1))

# (3) Partition the measurements now available in MSlist:

MSlist<-mzagglom(MSlist,dmzgap=10,ppm=TRUE,drtgap=500,minpeak=4,maxint=1E7)

# (4) EIC clustering of the partitions now available in MSlist:

MSlist<-mzclust(MSlist,dmzdens=5,ppm=TRUE,drtdens=120,minpeak=4)

# (5) Peak picking within the EIC clusters now available in MSlist:

MSlist<-mzpick(MSlist, minpeak = 4, drtsmall = 50, drtfill = 10,  drttotal = 200, recurs = 4,
weight = 2, SB = 3, SN=2, minint = 1E4, maxint = 1e+07, ended = 2)

# (6) Export a peak list now available in MSlist:

writePeaklist(MSlist,"directory","filename")

# (7) View your partitioning / EIC clustering / peak picking results:

plotMSlist(MSlist,ppmbar=10);
######################################################

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

```
# Do above steps (1) to (5) in one wrap, then export a peak list:

MSlist<-enviPickwrap( filepath.mzML,
                      MSlevel=c(1),
                      dmzgap=10,
                      dmzdens=5,
                      ppm=TRUE,
                      drtgap=1000,
                      drtsmall=20,
                      drtdens=250,
                      drtfill=10,
                      drttotal=200,
                      minpeak=4,
                      recurs=10,
                      weight=2,
                      SB=3,
                      SN=2,
                      minint=10E4,
                      maxint=10E6,
                      ended=2,
                      progbar=TRUE
                       )

writePeaklist(MSlist,"directory","filename")
##################################################


## End(Not run)
```

enviPickbatch *Batch peak picking for several .mzXML input files*

Description

Given a folder of .mzXML input files and subsequent processing, .csv peak tables with picked peaks are written to an output folder

Usage

```
enviPickbatch(folderin, folderout, MSlevel=c(1), dmzgap=15, dmzdens=4,
ppm=TRUE, drtgap=500, drtsmall=20, drtdens=250, drtfill=10, drttotal=200,
minpeak=4, recurs=10, weight=2, SB=3, SN=2, minint=1E5, maxint=1E7,
ended=2, ion_mode=FALSE, progbar=FALSE)
```

Arguments

`folderin` Folder containing .mzXML input files

`folderout` Destination folder for .csv peak tables

`MSlevel` numeric 1 (MS) or 2 (MS-MS), `readMSdata`

dmzgap	m/z gap width for partitioning, mzpart
dmzdens	Maximum measurement deviation (+/-) of m/z within an EIC, mzclust
ppm	dmzgap, given in ppm (TRUE) or as absolute value (FALSE)
drtgap	RT gap width for partitioning, mzpart
drtsmall	Peak definition - RT window of a peak; cp. minpeak argument
drtdens	Maximum length of EICs, mzclust
drtfill	RT gap length to be filled, mzpick
drttotal	Maximum RT length of a single peak, mzpick
minpeak	Peak definition - minimum number of measurements per peak to found in windows of drtsmall and dmzsmall, mzpick
recurs	Maximum number of peaks within one EIC, mzpick
weight	Weight for assigning measurements to a peak, mzpick
SB	Signal-to-base ratio, mzpick
SN	Signal-to-noise ratio, mzpick
minint	Minimum intensity of a peak, mzpick
maxint	Peaks above this intensity are alwas included, regardless of checks for SB or SN, mzpick
ended	Within the peak detection recursion set by argument recurs, how often can a peak detection fail, to end the recursion? mzpick
ion_mode	"positive" or "negative" ionization. Otherwise set to FALSE, see details.
progbar	Show a progress bar (TRUE or FALSE)? May only work under Windows OS.

Details

For further details on the parameter settings, please refer to the arguments of the underlying functions readMSdata, mzagglom, mzclust and mzpick. ion_mode allows to filter scans of a specific polarity from .mzXML files, if not set to FALSE; useful for files acquired under polarity switching.

Value

Folder with .csv peak tables, each containing columns with: m/z (mean m/z of peak measurements), var_m/z (m/z variation of peak measurements), max_int (base-line corrected maximum intensity), sum_int (sum of all base-line corrected peak measurement intensities), RT (retention time at maximum intensity), minRT (start peak RT), maxRT (end peak RT), peak# (peak ID number), EIC# (EIC ID number), Score (not yet implemented)

Author(s)

Martin Loos

See Also

readMSdata mzagglom mzclust mzpick plotMSlist

enviPickwrap *Peak detection for LC-HRMS .mzML and .mzXML data*

Description

A wrapper combining data upload, partitioning, EIC clustering and EIC peak detection from readMSdata, mzagglom, mzclust and mzpick.

Usage

```
enviPickwrap(filepath.mzXML, MSlevel=c(1), dmzgap=15, dmzdens=4, ppm=TRUE,
drtgap=500, drtsmall=20, drtdens=250, drtfill=10, drttotal=200, minpeak=4,
recurs=3, weight=2, SB=3, SN=2, minint=1E5, maxint=1E7, ended=2,
ion_mode=FALSE, progbar=FALSE)
```

Arguments

filepath.mzXML Path to the .mzXML file to be read, readMSdata

MSlevel numeric 1 (MS) or 2 (MS-MS), readMSdata

dmzgap m/z gap width for partitioning, mzpart

dmzdens Maximum measurement deviation (+/-) of m/z within an EIC, mzclust

ppm dmzgap, given in ppm (TRUE) or as absolute value (FALSE)

drtgap RT gap width for partitioning, mzpart

drtsmall Peak definition - RT window of a peak; cp. minpeak argument

drtdens Maximum length of EICs, mzclust

drtfill RT gap length to be filled, mzpick

drttotal Maximum RT length of a single peak, mzpick

minpeak Peak definition - minimum number of measurements per peak to found in windows of drtsmall and dmzsmall, mzpick

recurs Maximum number of peaks within one EIC, mzpick

weight Weight for assigning measurements to a peak, mzpick

SB Signal-to-base ratio, mzpick

SN Signal-to-noise ratio, mzpick

minint Minimum intensity of a peak, mzpick

maxint Peaks above this intensity are alwas included, regardless of checks for SB or SN,mzpick

ended Within the peak detection recursion set by argument recurs, how often can a peak detection fail, to end the recursion? mzpick

ion_mode "positive" or "negative" ionization. Otherwise set to FALSE, see details.

progbar Show a progress bar (TRUE or FALSE)? May only work under Windows OS.

Details

For further details on the parameter settings, please refer to the arguments of the underlying functions `readMSdata`, `mzagglom`, `mzclust` and `mzpick`. `ion_mode` allows to filter scans of a specific polarity from .mzXML files, if not set to FALSE; useful for files acquired under polarity switching.

Value

MSlist

`State` MSlist[[1]]: tags the individual steps the MSlist has undergone for peak picking.

`Parameters` MSlist[[2]]: saves the parameter settings.

`Results` MSlist[[3]]: saves the result summary values

`Scans` MSlist[[4]]: matrix with raw measurements (m/z, intensity, RT) and tags for partitions, EIC cluster or individual peaks.

`Partition_Index` MSlist[[5]]: Index assigning partitions to sections in the raw measurment of MSlist[[4]]. Required for fast (random) access, e.g., plotting.

`EIC_index` MSlist[[6]]: Index assigning EIC clusters to sections in the raw measurment of MSlist[[4]]. Required for fast access.

`Peak_index` MSlist[[7]]: Index assigning picked peaks to sections in the raw measurment of MSlist[[4]]. Required for fast access.

`Peaklist` MSlist[[8]]: Final peak list, cp. `mzpick`

Author(s)

Martin Loos

See Also

`readMSdata` `mzagglom` `mzclust` `mzpick` `plotMSlist`

`mzagglom` *Agglomerative partitioning of raw LC-HRMS measurements*

Description

Agglomerative partitioning of LC-HRMS measurements. Preparatory step for `mzclust` and `mzpick`. Requires an MSlist initilialized by `readMSdata` as input.

Usage

```
mzagglom(MSlist, dmzgap = 10,  ppm = TRUE, drtgap = 500, minpeak = 4,
maxint=1E7, progbar=FALSE)
```

Arguments

MSlist	MSlist generated by readMSdata
dmzgap	m/z gap width for partitioning
ppm	dmzgap given in ppm (TRUE) or as absolute value (FALSE)?
drtgap	RT gap width for partitioning
minpeak	Minimum number of measurements in a partition
maxint	Measurements equal or above this intensity will be retained even if ranging below minpeak
progbar	For debugging, ignore

Details

Partitioning of the full set of measurements into subsets is necessary to speed up the clustering procedure of mzclust. To this end, an agglomerative partitioning approach is used, combining measurements that are linked by values smaller than drtgap and dmzgap into single subsets. No measurements of two different subsets can be closer than drtgap and dmzgap to each other.

Value

Returns the argument MSlist, with entries made:

Parameters	MSlist[[2]]: saves the parameter settings.
Scans	MSlist[[4]]: matrix with raw measurements and tags resorted for partitions.
Partition_Index	MSlist[[5]]: Index assigning partitions to sections in the raw measurement of MSlist[[4]]; required for fast (random) access.

Imbecile

Do not set minpeak bigger than its counterpart in mzclust or mzpick. Too complicated? Then rather use enviPickwrap for adjusting all function arguments.

Warning

Despite optimized code, this function has a potential to run for a intolerable long time or out of memory if (a) the parameters are set wrongly, (b) the .mzML/.mzXML-file was not centroided or (c) the underlying data is inadequate for this peak picker. With regards to (a), do not assume gaps being larger than actually present. Instead, use plotMSlist to have a look at your data contained in MSlist after upload with readMSdata.

Author(s)

Martin Loos

See Also

mzclust,mzpart

mzclust *EIC-clustering of LC-HRMS data.*

Description

Based on the measurement partitions generated by mzagglom, extracted ion chromatograms (EICs) are assigned by a clustering procedure. Preparatory step for mzpick.

Usage

```
mzclust(MSlist,dmzdens=10,ppm=TRUE,drtdens=60,minpeak=4,maxint=1E6,
progbar=FALSE,merged=TRUE,from=FALSE,to=FALSE   )
```

Arguments

`MSlist` MSlist returned by mzpart

`dmzdens` Maximum measurement deviation (+/-) of m/z from its mean within each EIC

`ppm` `dmzdens` given in ppm (TRUE) or as absolute value (FALSE)?

`drtdens` Retention time (RT) tolerance for clustering; defined as (+/-) time units relative to the lowest and highest RT value in each cluster

`minpeak` Minimum number of measurements expected in an EIC

`maxint` EIC cluster with measurements above this intensity are kept, even if they do not fulfill `minpeak`

`progbar` For debugging, ignore

`merged` Merge EIC cluster of comparable m/z (TRUE or FALSE)?

`from` For debugging, ignore

`to` For debugging, ignore

Details

Within individual partitions calculated by mzagglom, an unsupervised clustering of measurements to individual ion chromatograms (EICs) is performed. For this purpose, a first EIC cluster is initialized with the most intense measurement, given an m/z uncertainty of 2*`dmzdens`. Along decreasing intensities, all other measurements are then sequentially either assigned to this cluster or used to define new clusters. For assignment, measurements must range both within the current tolerances of `dmzdens` and `drtdens` of an existing cluster. If several cluster are eligible for assignment, the one with the smallest mass difference between measurement m/z and cluster mean m/z will be used. Each time a new assignment to an existing cluster is made, its m/z estimate can be improved, i.e., the `dmzdens` tolerance around its mean m/z gradually shrinks from 2*`dmzdens` to `dmzdens`. In addition, `dmzdens` is used to update the RT tolerance of a cluster at each assignment.

With no measurements left, EIC clusters nested in m/z are then merged, relative to the m/z boundaries of the most intense cluster and stepwise along increasing mean m/z differences. Finally, EIC cluster are filtered to fulfill either `minpeak` or `maxint`.

Value

Returns the argument MSlist, with entries made:

`Parameters`	MSlist[[2]]: saves the parameter settings.
`Scans`	MSlist[[4]]: matrix with raw measurements and tags resorted for EIC clusters within the partition subsets.
`EIC_index`	MSlist[[6]]: Index assigning EIC cluster to sections in the raw measurement of MSlist[[4]]; required for fast (random) access.

Note

Too small values for `dmzdens` or too large `drtdens` may cause erratic EICs cluster

Author(s)

Martin Loos

References

Loos, M. (XXXX). Extraction of ion chromatograms by unsupervised clustering of high-resolution mass spectrometry data. Some Journal. Sometime.

See Also

`mzpick`

`mzpart` *Divisive partitioning of raw LC-HRMS measurements*

Description

Divisive recursive partition of LC-HRMS measurements. Preparatory step for `mzclust` and `mzpick`; altenative to `mzagglom`. Requires an MSlist initilialized by `readMSdata` as input.

Usage

```
mzpart(MSlist, dmzgap = 10, drtgap = 500, ppm = TRUE,
minpeak = 4, peaklimit = 2500, cutfrac = 0.1, drtsmall=50,
progbar = FALSE, stoppoints = 2e+05)
```

Arguments

MSlist	MSlist generated by readMSdata
dmzgap	m/z gap width for partitioning
drtgap	RT gap width for partitioning
ppm	dmzgap given in ppm (TRUE) or as absolute value (FALSE)?
minpeak	Minimum number of measurements in a partition
peaklimit	Maximum number of measurements in a partition
cutfrac	Fraction of low density measurements to be discarded
drtsmall	RT tolerance used to estimate density
progbar	For debugging, ignore
stoppoints	For debugging, ignore

Details

This function searchs recursively for gaps in retention time (RT) and m/z in the LC-HRMS measurements and thus partitions (and resorts) the matrix contained in MSlist[[4]]. If neither partitioning by RT nor by m/z results in a small enough partition of <= peaklimit measurements, a fraction cutfrac of lowest-density measurements is discarded and the partition procedure resumed. Measurement-wise density is based on a gaussian kernel density estimate scaled to dmzgap and drtsmall, i.e., to the local neighbourhood of each measurement.

Partitioning is necessary to speed up the clustering procedure of mzclust. Hence, there is a tradeoff: large values of peaklimit leads to faster execution of mzpart but to slower computation of mzclust and vice versa.

Value

Returns the argument MSlist, with entries made:

Parameters	MSlist[[2]]: saves the parameter settings.
Scans	MSlist[[4]]: matrix with raw measurements and tags resorted for partitions.
Partition_Index	MSlist[[5]]: Index assigning partitions to sections in the raw measurement of MSlist[[4]]; required for fast (random) access.

Imbecile

Do not set minpeak bigger than its counterpart in mzclust or mzpick. Too complicated? Then rather use enviPickwrap for adjusting all function arguments.

Warning

Despite optimized code, this function has a potential to run for a intolerable long time or out of memory if (a) the parameters are set wrongly, (b) the .mzML/.mzXML-file was not centroided or (c) the underlying data is inadequate for this peak picker. With regards to (a), do not assume gaps being larger than actually present. Instead, use plotMSlist to have a look at your data contained in

MSlist after upload with readMSdata; set progbar=TRUE to monitor where a function fails. Once settled, set progbar=FALSE for faster execution.

To avoid running out of memory, stoppoints sets the maximum number of measurements that can be handled in the routines to delete those of lowest intensity (in cases where peaklimit cannot be reached by partitioning by dmzgap and drtgap alone). If above stoppoints, execution aborts.

Author(s)

Martin Loos

See Also

mzclust

mzpick *EIC peak picking*

Description

Peak-picking within individual EIC cluster formed by mzclust without assuming a certain peak shape. Includes a baseline subtraction step.

Usage

```
mzpick(MSlist, minpeak = 4, drtsmall = 20, drtfill = 10,  drttotal = 200, recurs = 4,
weight = 2, SB = 3, SN=2, minint = 1E4, maxint = 1e+07, ended = 2, progbar = FALSE,
from = FALSE, to = FALSE)
```

Arguments

MSlist	An MSlist returned by mzclust
minpeak	Peak definition - minimum number of measurements required within the RT window of drtsmall
drtsmall	Peak definition - RT window of a peak; cp. minpeak argument
drtfill	Maximum RT gap length to be filled, cp. details
drttotal	Peak definition - Maximum RT length of a single peak
recurs	Maximum number of peaks within one EIC, cp. details
weight	Weight for assigning measurements to a peak, cp. details
SB	Peak definition - signal-to-base ratio
SN	Peak definition - signal-to-noise ratio
minint	Peak definition - minimum intensity of a peak
maxint	Peaks above this intensity are always retained, regardless of other checks
ended	Within the peak detection recursion set by argument recurs, how often can a peak detection fail to end the recursion?

`progbar` For debugging, ignore

`from` For debugging, ignore

`to` For debugging, ignore

Details

In a first step, RT gaps between measurements in an EIC not larger than `drtfill` are filled by linear interpolation. Subsequently, peaks are assigned over a number of `recurs` recursions not interrupted by more than `ended` times of failed peak detections.

At each recursion, the most intense EIC measurement not yet assigned to a peak is selected as peak apex and neighbouring unassigned measurements at lower and higher RT are evaluated for forming the peak. To this end, increases (lower RT) and decreases (higher RT) in intensity of consecutive measurements over a maximum RT width of `drtdens` are summed and penalized by a factor of `weight` for intensity reversions. The measurements with optimum values are then selected to define the start and end measurement of the peak.

Thereupon, the candidate peak is checked to (a) have at least `minpeaks` within a RT window of `drtsmall`, (b) be larger than the minimum peak intensity `minint` and (c) have a minimum `SB` ratio (the ratio between the most intense measurement and the mimimum intensity of the first or last peak measurement). Candidate peaks failing in any of the aspects (a) to (c) are discarded (adding to `ended`), unless they are higher in intensity than `maxint`.

Next, all measurements assigned to peaks are removed from the EIC and the resulting gaps linearly interpolated and smoothed by a moving window average to form a baseline. The latter is then subtracted from the assigned peaks.

In a last step, peaks are checked for their signal-to-noise `SN` ratio in relation to the baseline measurements (if present). Herein, `SN` is defined as the ratio between the most intense (baseline-corrected) peak measurement and the median of the difference between the non-peak measurements (if any) and the baseline.

Value

Returns the argument MSlist, with entries made:

`Parameters` MSlist[[2]]: saves the parameter settings.

`Scans` MSlist[[4]]: matrix with raw measurements and tags for picked peaks within EICs within partitions.

`Peak_index` MSlist[[7]]: Index assigning picked peaks to sections in the raw measurment of MSlist[[4]]. Required for fast access.

`Peaklist` MSlist[[8]]: matrix with picked peak characteristics, with columns: m/z (mean m/z of peak measurements), var_m/z (m/z variation of peak measurements), max_int (base-line corrected maximum intensity), sum_int (sum of all base-line corrected peak measurement intensities), RT (retention time at maximum intensity), minRT (start peak RT), maxRT (end peak RT), peak# (peak ID number), EIC# (EIC ID number) and Score (not yet implemented).

imbecile

`ended` must be smaller than `recurs`. `minpeak` and `drtsmall` should be congruent in both and `mzclust` and `mzpick`.

Author(s)

Martin Loos

See Also

`plotMSlist` `writePeaklist`

`plotMSlist` *Interactive raw data and result viewer*

Description

View your centroided LC-HRMS data and partitioning / clustering / peak-picking results. Monitor what peak-picking produces and if it fails; get a grip on optimal parameter settings from comparison with underlying raw data.

Usage

```
plotMSlist(MSlist, RTlimit = FALSE, mzlimit = FALSE, shiny = FALSE, ppmbar = 8)
```

Arguments

`MSlist`	An MSlist returned by `readMSdata`, `mzpart`, `mzclust` or `mzpick`
`RTlimit`	Initialize plot: two-element vector of lower and upper RT plot limits. Set to FALSE to view full data.
`mzlimit`	Initialize plot: two-element vector of lower and upper m/z limits. Set to FALSE to view full data.
`shiny`	For debugging. Ignore.
`ppmbar`	Size of m/z bar (in ppm) shown at large zoom

Details

For more help, use the help button in the interactive plot. Based on low-level R plot functionality!

Note

`MSlist` may contain a lot of data; rendering of measurements for plotting may thus decrease in speed when zooming out or using the full-view mode.

Author(s)

Martin Loos

See Also

`readMSdata` `mzagglom` `mzclust` `mzpick`

readMSdata *Read LC-HRMS measurement .mzXML data*

Description

Initiates an MSlist object and reads LC-HRMS measurement data from .mzXML files.

Usage

```
readMSdata(filepath.mzXML, MSlevel=c(1), progbar=FALSE, minRT=FALSE,
maxRT=FALSE, minmz=FALSE, maxmz=FALSE, ion_mode=FALSE)
```

Arguments

`filepath.mzXML`	Path to the .mzXML file to be read
`MSlevel`	numeric 1 (MS) or 2 (MS-MS)
`progbar`	Show a progress bar (TRUE or FALSE)? Might only work in Windows OS
`minRT`	Filter for measurements with retention time >= minRT. Otherwise set to FALSE.
`maxRT`	Filter for measurements with retention time <= maxRT. Otherwise set to FALSE.
`minmz`	Filter for measurements with m/z >= minmz. Otherwise set to FALSE.
`maxmz`	Filter for measurements with m/z <= maxmz. Otherwise set to FALSE.
`ion_mode`	"positive" or "negative" ionization. Otherwise set to FALSE, see details.

Details

The return value, a so-called MSlist object, is a simple R list object that contains (a) the raw measurement data, (b) intermediate/final results of the peak picking procedure and (c) indices for random access, to be passed among functions. Peaks are nested in EIC clusters which in turn are nested in partitions which in turn are subsets of measurements; MSlist[[4]] is resorted accordingly during all peak picking steps.

Setting `minRT`, `maxRT`, `minmz` or `maxmz` allows you to filter your .mzML data. On the one hand, this may be very useful if only being interested in certain ranges of an experiment. On the other hand, this allows you to upload subset data of an experiment too large to be loaded into R at once. `ion_mode` allows to filter scans of a specific polarity from .mzXML files, if not set to FALSE; useful for files acquired under polarity switching.

Value

MSlist

`State`	MSlist[[1]]: tags the individual steps the MSlist has undergone.
`Parameters`	MSlist[[2]]: saves parameter settings.
`Results`	MSlist[[3]]: saves a result summary.
`Scans`	MSlist[[4]]: matrix with raw measurements (m/z, intensity, RT) and tags for partitions, EIC cluster and individual peaks.

`Partition_Index` MSlist[[5]]: Index assigning partitions to sections in the raw measurment of MSlist[[4]]. Needed for fast (random) access during, e.g., plotting.

`EIC_index` MSlist[[6]]: Index assigning EIC clusters to sections in the raw measurment of MSlist[[4]]. Required for fast access.

`Peak_index` MSlist[[7]]: Index assigning picked peaks to sections in the raw measurment of MSlist[[4]]. Required for fast access.

`Peaklist` MSlist[[8]]: Final peak list, cp. `mzpick`

Warning

Use `plotMSlist` to check your data in MSlist for consistency at an early stage before further processing.

Note

It is your responsibility to ensure your input files are centroided. If not, R may freeze and the peak picker will not return valid results.

Author(s)

Martin Loos

See Also

`mzagglom`

webpick *enviPick web browser UI*

Description

Run enviPick conveniently from a web browser-based user interface

Usage

```
webpick()
```

Arguments

check webpage

Warnings

May not work with Microsoft Internet Explorer; better choose a different default browser (e.g., Google Chrome).

Author(s)

Martin Loos

writePeaklist *Export a .csv peak table from an MSlist object*

Description

Given an MSlist object containing peak picking results from `mzpick`, export a peak table.csv.

Usage

```
writePeaklist(MSlist, directory, filename, overwrite = FALSE)
```

Arguments

`MSlist`	A MSlist object generated by `enviPickwrap` or `mzpick`
`directory`	Character string with the directory to write to
`filename`	Name of the .csv file to create
`overwrite`	TRUE/FALSE

Value

.csv table, with columns:

m/z (mean m/z of peak measurements), var_m/z (m/z variation of peak measurements), max_int (base-line corrected maximum intensity), sum_int (sum of all base-line corrected peak measurement intensities), RT (retention time at maximum intensity), minRT (start peak RT), maxRT (end peak RT), peak# (peak ID number), EIC# (EIC ID number), Score (not yet implemented)

Author(s)

Martin Loos

Index