Package ‘erah’

July 11, 2018

Type Package
Title Automated Spectral Deconvolution, Alignment, and Metabolite Identification in GC/MS-Based Untargeted Metabolomics
Version 1.1.0
Date 2018-07-10
Author Xavier Domingo-Almenara [aut, cre, cph], Sara Samino [aut], Maria Vinaixa [aut], Alexandre Perera [aut, ths], Jesus Brezmes [aut, ths], Oscar Yanes [aut, ths]
Depends R (>= 2.10), Rcpp
Imports ncdf4, nnls, igraph, signal, quantreg, XML, methods
Suggests R.rsp, mzR
Maintainer Xavier Domingo-Almenara <xdomingo@scripps.edu>
Description Automated compound deconvolution, alignment across samples, and identification of metabolites by spectral library matching in Gas Chromatography - Mass spectrometry (GC-MS) untargeted metabolomics. Outputs a table with compound names, matching scores and the integrated area of the compound for each sample. Package implementation is described in Domingo-Almenara et al. (2016) <doi:10.1021/acsaanalchem.6b02927>.
License GPL (>= 2)
URL http://metabolomicsplatform.com/
Repository CRAN
LazyData yes
NeedsCompilation yes
VignetteBuilder R.rsp
Date/Publication 2018-07-11 05:00:07 UTC

R topics documented:

alignComp .................................................. 2
alignList .................................................. 3
cmplnfo ................................................... 4
computeRIerror .......................................... 5
**alignComp**

Alignment of compounds

Alignment of GC-MS deconvolved compounds

**Usage**

`alignComp(Experiment, alParameters, blocks.size=NULL)`

**Arguments**

- **Experiment**
  A `MetaboSet` S4 object containing the experiment data previously created by `newExp` and deconvolved by `deconvolveComp`.

- **alParameters**
  The software alignment parameters object previously created by `setAlPar`.

- **blocks.size**
  For experiment of more than 1000 samples, and depending on the computer, alignment can be conducted by block segmentation. See details.
Details

See eRah vignette for more details. To open the vignette, execute the following code in R: vignette("eRahManual", package="erah")

For experiments containing more than 100 (Windows) or 1000 (Mac or Linux) samples (numbers depending on the computer resources and sample type). In those cases alignment can be conducted by block segmentation. For an experiment of e.g. 1000 samples, the block.size can be set to 100, so the alignment will perform as multiple (ten) 100-samples experiments, to later align them into a single experiment.

This parameter is designed to solve the typical problem that appear when aligning under Windows operating system: "Error: cannot allocate vector of size XX Gb". Such a problem will not appear with Mac or Linux, but several hours of computation are expected when aligning a large number of samples. Using block segmentation provides a greatly improved run-time performance.

Value

The function returns an updated S4 'MetaboSet' class, where the GC-MS samples have been now aligned.

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


See Also

newExp, setDecPar, deconvolveComp

Usage

alignList(object, by.area=TRUE)
Arguments

object  A 'MetaboSet' S4 object containing the experiment data. The experiment has to be previously deconvolved, aligned and (optionally) identified.

by.area  if TRUE (default), eRah outputs quantification by the area of the deconvolved chromatographic peak of each compound. If FALSE, eRah outputs the intensity of the deconvolved chromatographic peak.

Details

Returns an alignment table containing the list of aligned metabolites and their relative quantification for each sample in a given experiment.

Value

alignList returns a data frame object:

| AlignID | The unique Tag for found metabolite by eRah. Each metabolite found by eRah for a given experiment has an unique AlignID tag number. |
| Factor | the Factor tag name. Each metabolite has an unique 'Factor' name to enhance visual interpretation. |
| tmean | The mean compound retention time. |
| FoundIn | The number of samples in which the compound has been detected (the number of samples where the compound area is non-zero). |
| Quantification | As many columns as samples and as many rows as metabolites, where each column name has the name of each sample. |

See Also

idList, dataList

compInfo  Information of a Compound

Description

Displays basic information of a compound in the MS library.

Usage

compInfo(comp.id, id.database=mslib)

Arguments

comp.id  The DB.Id number of the compound.

id.database  The mass-spectra library to be compared with the empirical spectra. By default, the MassBank - Mass Bank of North America (MoNa) database are employed (mslib object).
computeRlerror

Details

Returns details on a given compound such as the synonyms, CAS, KEGG, retention index, among others.

See Also

findcomp

Examples

# finding proline
findComp("proline")

# we see that proline 2TMS has the DB.Id number 42, then:
compInfo(42)

computeRlerror Retention Index Error Computation

Description

Computes the retention indexes (RI) errors given an internal or external calibration curve

Usage

computeRlerror(experiment, id.database=mslib, reference.list, ri.error.type=c('relative','absolute'), plot.results=TRUE)

Arguments

Experiment A 'MetaboSet' S4 object containing the experiment data. The experiment has to be previously deconvolved, aligned and identified.

id.database The mass-spectra library to be compared with the empirical spectra. By default, the MassBank - Mass Bank of North America (MoNa) database are employed (mslib object).

reference.list The list containing the reference data to create the calibration curve (Internal or external calibration). Please see the Vignette and Examples.

ri.error.type Specify wether absolute or relative (default) RI error is to be computed.

plot.results Shows the RI/RT graphic.

Value

Returns an 'Experiment' object. To visualize the output use: idList() function.
Author(s)

Xavier Domingo-Almenara. xdomingo@scripps.edu

References


See Also

idList, dataList

Examples

#Option A: (The RT and RI of an external calibration is provided)

ex <- computeRerror(ex, mslib, reference.list=list(RT=c(4.4683, 7.4402, 8.8121, 11.5103), RI=c(1081.68, 1251.31, 1346.8, 1456.8)))
id.list <- idList(ex)

#Option B: (The AlignID of internally identified metabolites by eRah are provided, and used as a reference.)

ex <- computeRerror(ex, mslib, reference.list=list(AlignID = c(45,67,92,120)))
id.list <- idList(ex)

#Please, see Vignette for extended details, usage and advices!

Creating Experiment Tables

Description

eRah requires a instrumental and (optionally) phenotype .csv file for starting/creating a new eRah project/experiment. This function automatically creates the Phenotype and Instrumental data .csv files.

Usage

createdt(path)

Arguments

path the path where the experiment-folder is (where the experiment samples are stored).
The experiment has to been organized as follows: all the samples related to each class have to be stored in the same folder (one folder = one class), and all the class-folders in one folder, which is the experiment folder.

Two things have to be considered at this step: .csv files are different when created by American and European computers, so errors may raise due to that fact. Also, the folder containing the samples, must contain only folders. If the folder contains files (for example, already created .csv files), eRah will prompt an error.

See eRah vignette for more details. To open the vignette, execute the following code in R: vignette("eRahManual", package="erah")

**See Also**

newExp

**Examples**

```r
# Store all the raw data files in one different folder per class, 
# and all the class-folders in one folder, which is the experiment 
# folder. Then execute

createdt(path)

# where path is the experiment folder path.
# The experiment can be now startd by:

ex <- newExp(instrumental="path/DEMO_inst.csv", 
phenotype="path/DEMO_pheno.csv", info="DEMO Experiment")
```

dataList (Data list)

**Description**

The final eRah list of aligned and identified metabolites and their relative quantification for each sample in a given experiment

**Usage**

dataList(Experiment, id.database=mslib, by.area=TRUE)
Arguments

- **Experiment**: A `MetaboSet` S4 object containing the experiment data. The experiment has to be previously deconvolved, aligned and identified.

- **id.database**: The mass-spectra library to be compared with the empirical spectra. By default, the MassBank - Mass Bank of North America (MoNa) database are employed (mslib object).

- **by.area**: if TRUE (default), eRah outputs quantification by the area of the deconvolved chromatographic peak of each compound. If FALSE, eRah outputs the intensity of the deconvolved chromatographic peak.

Details

Returns an identification and alignment table containing the list of aligned and identified metabolites (names) and their relative quantification for each sample in a given experiment.

Value

alignList returns an S3 object:

- **AlignID**: The unique Tag for found metabolite by eRah. Each metabolite found by eRah for a given experiment has an unique AlignID tag number.

- **tmean**: The mean compound retention time.

- **FoundIn**: The number of samples in which the compound has been detected (the number of samples where the compound area is non-zero).

- **Name.X**: the name of the Xst/nd/rd... hit. idList return as many X (hits) as n.putative selected with `identifyComp`.

- **MatchFactor.X**: The match factor/score of spectral similarity (spectral correlation).

- **DB.Id.X**: The identification number of the library. Each metabolite in the reference library has a different DB.Id number.

- **CAS.X**: the CAS number of each identified metabolite.

- **Quantification**: As many columns as samples and as many rows as metabolites, where each column name has the name of each sample.

See Also

- `idList`, `alignList`
Deconvolution of compounds in samples

Description

Deconvolution of GC-MS data

Usage

deconvolveComp(experiment, decparameters, samples.to.process=NULL, down.sample=FALSE, virtual.scans.ps=NULL)

Arguments

- **experiment** A 'MetaboSet' S4 object containing the experiment data previously created by newExp.
- **decparameters** The software deconvolution parameters object previously created by setDecPar
- **samples.to.process** Vector indicating which samples are to be processed.
- **down.sample** If TRUE, chromatograms are down sampled to define one peak with 10 scan points (according to the minimum peak width). This is to process longer chromatograms with wider peak widths (more than 20 seconds peak width and small scans per second values). See details.
- **virtual.scans.ps** Manually correction of scans per second. When chromatograms are downsampled (too few scans per second, or too many), a virtual scans per second can be defined, and data interpolation will correct the data. However, reanalysis of experimental data is advised.

Details

See eRah vignette for more details. To open the vignette, execute the following code in R: vignette("eRahManual", package="erah")

eRah uses multivariate methods which run-time performance depend on the amount of data to be analyzed. When peaks are wide and the scans per second used for acquisition is too large, the number of points (scans) that define a peak might be too many, leading eRah to a poor run-time performance. To solve that, use down.sample=TRUE to allow eRah to define a peak with 10 seconds and analyze the data more efficiently

Value

The function returns an updated S4 'MetaboSet' class, where the GC-MS samples have been now deconvolved.
Author(s)
Xavier Domingo-Almenara. xavier.domingo@urv.cat

References

See Also
newExp, setAlPar

Examples

# Deconvolve data from a created experiment by newExp().
ex <- newExp(instrumental="path")

# The following will set eRah for analyzing the chromatograms
# from minutes 5 to 15, and without taking into account the masses
# 35:69,73:75,147:149, with a minimum peak width of 0.7 seconds.

ex.dec.par <- setDecPar(min.peak.width=0.7, min.peak.height=5000,
noise.threshold=500, avoid.processing.mz=c(35:69,73:75,147:149),
analysis.time=c(5,15))

# An now deconvolve the compounds in the samples:
ex <- deconvolveComp(ex, decParameters=ex.dec.par)

eRah_DB-class

Description
The eRah_DB class contains the slots for storing and accessing a MS library.

Slots
name: The name of the stored library
version: The version of the stored library (and which is the database identifier, should be unique and used to check if is the database used in other experiments)
info: Character vector containing complementary information about the library.
database: A list of S3 objects, which each object contains the information on a different compound.
Author(s)
Xavier Domingo-Almenara.

expClasses

Description
The classes of a given experiment.

Usage
expClasses(object)

Arguments
object A 'MetaboSet' S4 object containing the experiment.

Details
Returns the classes details of the experiment.

See Also
metadata, phenodata

export2CEF

Description
Export spectra to CEF format for comparison with the NIST library through MassHunter interface.

Usage
export2CEF(Experiment, export.id=NULL, id.database = mslib, store.path=getwd())

Arguments
Experiment A 'MetaboSet' S4 object containing the experiment.
export.id If NULL, all the spectra in the experiment will be exported. Otherwise, only the AlignID in export.id will be exported.
id.database The mass-spectra library used in the experiment.
store.path The path where the converted files are to be exported.
**export2MSP**  
*Export spectra to MSP*

**Description**
Export spectra to MSP format for comparison with the NIST library.

**Usage**
```r
export2MSP(Experiment, export.id=NULL,  
  id.database = mslib, store.path=getwd(),  
  alg.version=1)
```

**Arguments**
- **Experiment**: A `MetaboSet` S4 object containing the experiment.
- **export.id**: If NULL, all the spectra in the experiment will be exported. Otherwise, only the AlignID in export.id will be exported.
- **id.database**: The mass-spectra library used in the experiment.
- **store.path**: The path where the converted files are to be exported.
- **alg.version**: Different algorithm implementations. Users have to chose what version works with their NIST MSearch or other software version. By default, alg.version is set to 1. If it not works, try setting alg.version to 2 ;).

---

**findComp**  
*Find a Compound*

**Description**
Finds compounds in the MS library by Name, CAS or chemical formula.

**Usage**
```r
findComp(name = NULL, id.database = mslib, CAS = NULL,  
  chem.form = NULL)
```

**Arguments**
- **name**: The name of the compound to be found.
- **id.database**: The mass-spectra library to be compared with the empirical spectra. By default, the MassBank - Mass Bank of North America (MoNa) database are employed (mslib object).
- **CAS**: The CAS number of the compound to be found.
- **chem.form**: The chemical formula of the compound to be found.
Value
findComp returns an S3 object:

<table>
<thead>
<tr>
<th>DB.Id</th>
<th>The identification number of the library. Each metabolite in the reference library has a different DB.Id number.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compound Name</td>
<td>Compound Name.</td>
</tr>
<tr>
<td>CAS</td>
<td>CAS number</td>
</tr>
<tr>
<td>Formula</td>
<td>Chemical Formula.</td>
</tr>
</tbody>
</table>

See Also
compInfo

Examples

```r
# finding proline
findComp("proline")

# be careful, exact matches are not supported, # as well as special cases of partial searches, like these cases:
findComp("L-proline (2TMS)")
findComp("proline 2")

# which will not report any results despite being in the database
```

identifyComp | Identification of compounds

Description
Identification of compounds. Each empirical spectrum is compared against a ms library.

Usage
identifyComp(Experiment, id.database=mslib, 
mz.range=NULL, n.putative=3)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment</td>
<td>A 'MetaboSet' S4 object containing the experiment data previously created by newExp, deconvolved by deconvolveComp and optionally aligned by align-Comp.</td>
</tr>
</tbody>
</table>
idList

id.database  The mass-spectra library to be compared with the empirical spectra. By default, the MassBank-[2] - Mass Bank of North America (MoNa) database are employed.

mz.range  The same as in alignComp. If specified already in alignComp, then there is no need to specify it again. If not, it has to be specified.

n.putative  The number of hits (compound candidate names) to be returned for each spectrum found.

Value

The function returns an updated S4 'MetaboSet' class, where the GC-MS samples have been now aligned.

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


See Also

newExp, alignComp, setAlPar, setDecPar

idList  Identification list

Description

The list of identified metabolites in a given experiment

Usage

idList(object, id.database=mslib)
importGMD

Arguments

object A 'MetaboSet' S4 object containing the experiment data. The experiment has to be previously deconvolved, aligned and identified.

id.database The mass-spectra library to be compared with the empirical spectra. By default, the MassBank - Mass Bank of North America (MoNa) database are employed (mslib object).

Details

Returns an identification table containing the names, match scores, and other variables for a given experiment.

Value

idList returns an S3 object:

| AlignID | The unique Tag for found metabolite by eRah. Each metabolite found by eRah for a given experiment has an unique AlignID tag number. |
| tmean   | The mean compound retention time. |
| Name.X  | the name of the Xst/nd/rd... hit. idList return as many X (hits) as n.putative selected with identifyComp. |
| FoundIn | The number of samples in which the compound has been detected (the number of samples where the compound area is non-zero). |
| MatchFactor.X | The match factor/score of spectral similarity (spectral correlation). |
| DB.Id.X | The identification number of the library. Each metabolite in the reference library has a different DB.Id number. |
| CAS.X   | the CAS number of each identified metabolite. |

See Also

alignList, dataList

---

importGMD Import MSP files from GMD to R

Description

Import the Golm Metabolome Database.

Usage

importGMD(filename, DB.name, DB.version, DB.info, type=c("VAR5.ALK","VAR5.FAME","MDN35.ALK","MDN35.FAME"))
importMSP

Arguments

filename The filepath containing the GMD database file.
DB.name The name of the database (each user may chose its own name)
DB.version The version of the database (each user may chose its own version)
DB.info Some info about the database for further reference
type The type of RI to be imported from the database

Details

For more details, please see the eRah manual

---

importMSP Import MSP files to R

Description

Import MS libraries in MSP format to eRah DB format.

Usage

importMSP(filename, DB.name, DB.version, DB.info)

Arguments

filename The filepath containing the MSP library file.
DB.name The name of the database (each user may chose its own name)
DB.version The version of the database (each user may chose its own version)
DB.info Some info about the database for further reference

Details

The MSP input file should look like:

Name: Metabolite_name
Formula: H2O
MW: 666
ExactMass: 666.266106
CAS#: 11-22-3
DB#: 1
Comments: Metabolite_name reference standard
Num Peaks: XX
53 1; 54 2; 55 5; 56 2; 57 2;
Name: Metabolite_name_2
Formula: H2O2
MW: 999
ExactMass: 999.266106
CAS#: 22-33-4
DB#: 2
Comments: Metabolite_name_"" reference standard
Num Peaks: XX
66 10; 67 1000; 155 560; 156 800; 157 2;
158 14; 159 1; 160 100; 161 2; 167 1;

OR

Name: Metabolite_name
Formula: H2O
MW: 666
ExactMass: 666.266106
CASNO: 11-22-3
DB#: 1
Comment: Metabolite_name reference standard
Num peaks: XX
53 1
54 2
55 5
Name: Metabolite_name_2
Formula: H2O2
MW: 999
ExactMass: 999.266106
CASNO: 22-33-4
DB#: 2
Comment: Metabolite_name_"" reference standard
Num Peaks: XX
66 10
67 1000
155 560

Or combinations of both.
For more details, please see the eRah manual.

---

**MetaboSet-class**  
*Class "MetaboSet"*

**Description**

The MetaboSet class is a single generic class valid for all sorts of metabolomic studies regardless of the experimental platform, the statistical processing and the annotation stage. It is the core operation class of eRah.

**Slots**

- **info**: Slot Info stores the general information of the experiment and the experimental platform used in the analysis of the biological samples.
- **data**: Slot Data contains either the raw data or the path of the files. It also contains the list of the selected features (deconvolved compounds). In the subslot Parameters it is saved the information regarding the feature selector algorithm (type, parameters, version...) and the experimental platform used.
- **metaData**: Slot MetaData has two slots. In the Instrumental slot it is saved a data frame with some mandatory fields (filename, date, time, sampleID) and optional fields related to the experimental platform (Column ID, Column Type, Ioniser,...). Slot Phenotypic contains a data frame with the sample and experimental information (phenotypes, longitudinal data,...).
- **results**: In the Results slot it is saved the information related to the statistical and identification results. The slot Parameters contains all the values of the parameters used in the identification and statistical functions. Slot Identification has the results of the identification process as well as the identification or/and annotation steps. The results of the statistical functions are saved in the Statistics slot.

**Author(s)**

Xavier Domingo-Almenara, Arnald Alonso and Francesc Fernandez-Albert.

---

**metaData**  
*Metadata*

**Description**

Displays the Experiment metadata.

**Usage**

`metaData(object)`
**mslib**

**Arguments**

- **object**
  A `MetaboSet` S4 object containing the experiment.

**See Also**

- `phenodata`

---

**mslib**

*MassBank Spectral Library*

**Description**

The default mass spectral library of eRah, which is the MassBank repository.

**Usage**

```r
data(mslib)
```

**Details**

This is the eRah default MS library, and automatically loaded with the eRah package. It contains almost 500 MS spectra. For details, see reference below.

**Author(s)**

The TOF-MS spectra were contributed by Kazusa DNA Research Institute, the Engineering Department of Osaka University and Plant Science Center of RIKEN.

MassBank (http://www.massbank.jp/)

**References**


**See Also**

- `compInfo`
newExp

New Experiment

Description

Sets a new experiment for eRah

Usage

newExp(instrumental, phenotype=NULL, info=character())

Arguments

instrumental The path where the instrumental .csv file is located.
phenotype (optional) The path where the phenotypic .csv file is located.
info Experiment description

Details

See eRah vignette for more details. To open the vignette, execute the following code in R: vignette("eRahManual", package="erah")

Value

newExp returns an S4 object of the class 'MetaboSet'.

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


See Also

createdt, setDecPar, setAlPar

Examples

# Store all the raw data files in one different folder per class,
# and all the class-folders in one folder, which is the experiment
# folder. Then execute

createdt(path)
# where path is the experiment folder path.
# The experiment can be now started by:

ex <- newExp(instrumental="path/DEM0_inst.csv",
phenotype="path/DEM0_pheno.csv", info="DEM0 Experiment")

---

### phenoData

**Show Phenotyphe data**

**Description**

Displays the Experiment phenotypic data (if included).

**Usage**

```
phenoData(object)
```

**Arguments**

- **object**
  - A 'MetaboSet' S4 object containing the experiment.

**See Also**

- metadata
- plotAlign

---

### plotAlign

**Plotting chromatophic profile with and without alignement**

**Description**

Plots the chromatophic profiles of the compounds found by eRah. Similarly to plotProfile, but with two sub-windows, showing the chromatophic profiles before and after alignment.

**Usage**

```
plotAlign(experiment, AlignId, per.class=T, xlim=NULL)
```

**Arguments**

- **experiment**
  - A 'MetaboSet' S4 object containing the experiment after being deconolved, aligned and (optionally) identified.
- **AlignId**
  - the Id identifier for the compound to be shown.
- **per.class**
  - logical. if TRUE the profiles are shown one color per class, if FALSE one color per sample.
- **xlim**
  - x axis (retention time) limits (see plot.default).
Author(s)
Xavier Domingo-Almenara. xavier.domingo@urv.cat

See Also
plotSpectra, plotProfile

plotChr

Plotting sample chromatogram

Description
Plot the sample chromatogram.

Usage
plotChr(experiment, nNsample=1, type=c("BIC","TIC","EIC"),
xlim=NULL, mz=NULL)

Arguments
- **Experiment**: A 'MetaboSet' S4 object containing the experiment.
- **nNsample**: Integer. The number of the sample to query.
- **type**: The type of plotting, Base Ion Chromatogram (BIC), Total Ion Chromatogram (TIC), or Extracted Ion Chromatogram (EIC).
- **xlim**: The range in minutes, separated by commas: c(rt.min, rt.max) of the limits of plotting. By default, all the chromatogram is plotted.
- **mz**: Just when EIC is selected. The range separated by commas: c(mz.min, mz.max) or a vector of numbers: c(50,67,80), of the masses to be plotted.

See Also
sampleInfo

Examples

# First, an experiment has to be already created by newExp()
# then, each sample chromatogram can be plotted by:

plotChr(Experiment, 1, "BIC")
plotChr(Experiment, 1, "TIC", xlim=c(5,7)) #Plots from minute 5 to 7.

plotChr(Experiment, 1, "EIC", mz=50:70, xlim=c(5,7)) #Plots
# from minute 5 to 7, and only the masses from 50 to 70.

plotChr(Experiment, 1, "EIC", xlim=c(7,7.5), mz=c(50,54,70)) #Plots
# the EIC from minute 7 to 7.5, and only the masses 50, 54 and 70.
plotProfile  

Plotting chromatophic profile

Description

Plots the chromatophic profiles of the compounds found by eRah.

Usage

plotProfile(experiment, AlignId, per.class=T, xlim=NULL)

Arguments

- **experiment**: A 'MetaboSet' S4 object containing the experiment after being deconvolved, aligned and (optionally) identified.
- **AlignId**: the Id identificator for the compound to be shown.
- **per.class**: logical. if TRUE the profiles are shown one color per class, if FALSE one color per sample.
- **xlim**: x axis (retention time) limits (see `plot.default`).

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

See Also

`plotSpectra`, `plotAlign`

plotSpectra  

Plotting spectra

Description

Plots the empirical spectra found by eRah, and allows comparing it with the reference spectra.

Usage

plotSpectra(Experiment, AlignId, n.putative=1, compare=T, id.database=mslib, comp.db=NULL, return.spectra=F, draw.color="purple", xlim=NULL)
plotSpectra

Arguments

- **Experiment**: A `MetaboSet` S4 object containing the experiment after being deconvolved, aligned and (optionally) identified.
- **AlignId**: the Id identifier for the compound to be shown.
- **n.putative**: The hit number (position) to be returned when comparing the empirical spectrum with the reference. See details.
- **compare**: logical. If TRUE, then the reference spectrum from the library is shown for comparison.
- **id.database**: The mass-spectra library to be compared with the empirical spectra. By default, the MassBank-[2] - Mass Bank of North America (MoNa) database are employed.
- **comp.db**: If you want to compare the empirical spectrum with another spectrum from the database, select the comp.db number from the database.
- **return.spectra**: logical. If TRUE, the function returns the empirical spectrum for the selected compound.
- **draw.color**: Selects the color for the reference spectrum (see colors).
- **xlim**: x axis (mass - m/z) limits (see plot.default).

Details

When identification is applied (see identifyComp), the number of hits to be returned (n.putative) has to be selected. Therefore, here you can compare the empirical spectrum (found by eRah) with each n.putative hit returned (1, 2, ...) by (see identifyComp).

Value

plotSpectra returns an vector when return.spectra=TRUE.

- **x**: vector. Contains the empirical spectrum.

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


RawDataParameters-class

See Also
plotProfile, plotAlign

Description
The RawDataParameters class contains the slots for storing and accessing into a MS sample, and the essential parameters for performing its processing (deconvolution).

Slots
- data: The data matrix of the sample to be processed
- min.mz: The minimum acquired mz number
- max.mz: The maximum acquired mz number
- start.time: Starting time of acquisition
- mz.resolution: Mz resolution
- scans.per.second: Scans per second
- avoid.processing.mz: Which mz do not have to be processed
- min.peak.width: Minimum peak width (stored in scans)
- min.peak.height: Minimum peak height
- noise.threshold: The noise threshold
- compression.coef: Compression coefficient (parameter for Orthogonal Signal Deconvolution)

Author(s)
Xavier Domingo-Almenara.

recMissComp Missing compound recovery

Description
Missing compounds recovery: fits a general model (all the compounds above a certain minimum number of samples) to all the samples.

Usage
recMissComp(Experiment, min.samples, free.model=F)
sampleInfo

Arguments

- **Experiment**: A 'MetaboSet' S4 object containing the experiment data previously created by `newExp`, deconvolved by `deconvolveComp` and aligned by `alignComp`.
- **min.samples**: The minimum number of samples in which a compound has to appear to be considered for searching into the rest of the samples where this compound missing.
- **free.model**: If TRUE, the spectra found in the samples where the compound is missing is used to get the final average spectra. (See details)

Details

**WARNING**: If compounds were previously identified, they have to be identified again after applying the "recMissComp" function. This means that "identifyComp" function has to be executed always after "recMissComp" for identification of compounds, even if "identifyComp" has been previously applied before.

The `free.model` parameter is recommended to be always FALSE (except for carbon tracking applications). This is because the spectra of the samples where the compound is missing is usually affected by noise, and this could decrease the matching score for a certain compound.

Value

The function returns an updated S4 'MetaboSet' class, where the GC-MS samples have been now aligned.

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


See Also

`newExp, alignComp, setAlPar, setDecPar`

---

**sampleInfo**  
*Information of the samples*

Description

Returns basic information on the samples.

Usage

`sampleInfo(Experiment, N.sample=1)`
setAlPar

Arguments

  Experiment A 'MetaboSet' S4 object containing the experiment.
  N.sample Integer. The number of the sample to query.

Details

  Returns details on a given sample of the experiment, such as name, start time, end time, minimum and maximum acquired m/z and scans per second.

See Also

  plotChr

setAlPar Set Alignment Parameters

Description

  Setting alignment parameters for eRah.

Usage

  setAlPar(min.spectra.cor, max.time.dist, 
            mz.range=c(70:600))

Arguments

  min.spectra.cor Minimum spectral correlation value. From 0 (non similar) to 1 (very similar). This value sets how similar two or more compounds have be to be considered for alignment between them.
  max.time.dist Maximum retention time distance. This value (in seconds) sets how far two or more compounds can be to be considered for alignment between them.
  mz.range The range of masses that is considered when comparing spectra.

Author(s)

  Xavier Domingo-Almenara. xavier.domingo@urv.cat

References

See Also

cnewExp, setDecPar, alignComp

Examples

# The following will set eRah for aligning compounds which are
# at least 90 (per cent) similar, and which peaks are at a
# maximum distance of 2 seconds. All the masses are considered when
# computing the spectral similarity.

ex.al.par <- setAlPar(min.spectra.cor=0.90, max.time.dist=2, mz.range=1:600)

setDecPar

Set Software Parameters

Description

Sets Software Parameters for eRah.

Usage

setDecPar(min.peak.width, min.peak.height=2500,
noise.threshold=500,
avoid.processing.mz=c(73:75,147:149),
compression.coef=2, analysis.time=0)

Arguments

min.peak.width  Minimum compound peak width (in seconds). This is a critical parameter that
conditions the efficiency of eRah. Typically, this should be the half of the mean
compound width.

min.peak.height  Minimum compound peak height

noise.threshold  Data below this threshold will be considered as noise

avoid.processing.mz  The masses that do not want to be considered for processing. Typically, in GC-
MS those masses are 73,74,75,147,148 and 149, since they are the ubiquitous mass fragments typically generated from compounds carrying a trimethylsilyl moiety.

compression.coef  Data is compressed when using the orthogonal signal deconvolution (OSD) algorithm according to this value. A level 2 of compression is recommended.

analysis.time  The chromatographic retention time window to process. If 0, all the chromatogram is processed.
Details

See eRah vignette for more details. To open the vignette, execute the following code in R: vignette("eRahManual", package="erah")

Author(s)

Xavier Domingo-Almenara. xavier.domingo@urv.cat

References


See Also

newExp, deconvolveComp, alignComp, setAlPar

Examples

# The following will set eRah for analyzing the chromatograms
# from minutes 5 to 15, and without taking into account the masses
# 35:69,73:75,147:149, with a minimum peak width of 0.7 seconds.

ex.dec.par <- setDecPar(min.peak.width=0.7, min.peak.height=5000,
noise.threshold=500, avoid.processing.mz=c(35:69,73:75,147:149),
analysis.time=c(5,15))

dec.par <- decConvolveComp(ex.dec.par)

dec.par <- decAlignComp(ex.dec.par)

dec.par <- decSetAlPar(ex.dec.par)

dec.par <- decAnalyze(ex.dec.par)

dec.par <- decQuantify(ex.dec.par)

dec.par <- decIdentify(ex.dec.par)

dec.par <- decExport(ex.dec.par)

show.MetaboSet

Show MetaboSet object

Description

Show MetaboSet object
Index

alignComp, 2, 14, 26, 28, 29
alignList, 3, 8, 15

colors, 24
compInfo, 4, 13, 19
computeRerror, 5
createdt, 6, 20
dataList, 4, 6, 7, 15
deconvolveComp, 3, 9, 29
eRah_DB-class, 10
expClasses, 11
export2CEF, 11
export2MSP, 12
findComp, 5, 12
identifyComp, 8, 13, 15, 24
idList, 4, 6, 8, 14
importGMD, 15
importMSP, 16
MetaboSet-class, 18
metaData, 11, 18, 21
mslib, 19
newExp, 3, 7, 10, 14, 20, 26, 28, 29
phenoData, 11, 19, 21
plot.default, 21, 23, 24
plotAlign, 21, 23, 25
plotChr, 22, 27
plotProfile, 22, 23, 25
plotSpectra, 22, 23, 23
RawDataParameters-class, 25
recMissComp, 25
sampleInfo, 22, 26
setAlPar, 10, 14, 20, 26, 27, 29
setDecPar, 3, 14, 20, 26, 28, 28
show.MetaboSet, 29