Package ‘iontree’
March 28, 2017

Type Package
Title Data management and analysis of ion trees from ion-trap mass spectrometry
Version 1.20.0
Depends methods, rJava, RSQLite, XML
Suggests iontreeData
Author Mingshu Cao
Maintainer Mingshu Cao <mingshu.cao@agresearch.co.nz>

Description Ion fragmentation provides structural information for metabolite identification. This package provides utility functions to manage and analyse MS2/MS3 fragmentation data from ion trap mass spectrometry. It was designed for high throughput metabolomics data with many biological samples and a large number of ion trees collected. Tests have been done with data from low-resolution mass spectrometry but could be readily extended to precursor ion based fragmentation data from high resolution mass spectrometry.

License GPL-2
biocViews Metabolomics, MassSpectrometry
LazyLoad yes
NeedsCompilation no

R topics documented:

 iontree-package ........................................... 2
 buildIonTree ........................................... 2
 createDB ............................................... 3
 distMS2 ............................................... 4
 formatSpec ........................................... 4
 getMetaInfo ........................................... 5
 getMSnRaw ........................................... 5
 hasMS2 ............................................... 6
 iontree-class ......................................... 7
 metaDataImport ...................................... 7
 mzImport .............................................. 8
 plot-methods ......................................... 9
 plotSpectrum ........................................ 9
 rs2iontree .......................................... 10
 saveMSnRaw ......................................... 10


**iontree-package**  
*MSn-iontree: Ion tree management and analysis*

**Description**
management and analysis of ion fragmentation data

**Details**

<table>
<thead>
<tr>
<th>Package:</th>
<th>iontree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>

This package provides functions to retrieve MSn fragmentation data, build MS2/MS3 ion trees from ion-trap low resolution mass spectrometry and to create a relational database (SQLite-based) for routine management of ion tree data. Other functions include metrics for MS2 spectral similarity measurement, iontree plotting and DB operations.

**Author(s)**

Mingshu Cao  
Maintainer: Mingshu Cao <mingshu.cao@agresearch.co.nz>

**References**

JRAP – a Java library was used for parsing mzXML and mzML  

---

**buildIonTree**  
*Build ion tree*

**Description**
build an ion tree derived from the specified m/z and RT ranges in one sample based on ms2 and ms3 raw data, see saveMSnRaw and hasMS2.

**Usage**

```r
buildIonTree(mzRange = c(340.5, 341.5), rtRange = c(270, 282), ms2, ms3)
```
createDB

Create a SQLite database

Description
create a relational database based on a schema defined in this package if argument sql is not specified. There are two tables (experiment and mz) defined to capture necessary information to annotate ions or peaks.

Usage
createDB(dbname = "mzDB.db", sql = "mzDBSchema.sql")

Arguments
- dbname: database name
- sql: predefined schema, or a modified definition

Value
A database file saved in the current folder
### Examples

```r
## Not run:
createDB(dbname="my.db")
## End(Not run)
```

---

### distMS2  
*Distance metric for MS2 spectral similarity measurement*

**Description**

distance metric for MS2 spectral comparison. MS2 spectrum is provided as 2-col matrix.

**Usage**

distMS2(a, b, topIon = 20)

**Arguments**

- `a`: MS2 spectrum
- `b`: MS2 spectrum
- `topIon`: the number of the most intense ions used for comparison

---

### formatSpec

*Format mass spec matrix data into a string format, or vice versa*

**Description**

argument `x` is a 2-column matrix of mz and intensity, or a string format of mz-intensity paris. Character pair of mz and intensity is separated by semicolon, for example, `150 2345.6; 151 4325.67; ....` which is often used to represent a mass spectrum as seen in NIST and MassBank.

**Usage**

formatSpec(x, fromTo = c("mat2str", "str2mat"))
**getMetaInfo**

**Arguments**

- **x**: 2-col matrix or type of character depends on "fromTo"
- **fromTo**: type of conversion

**Author(s)**

Mingshu Cao

**Examples**

```latex
x = "150 2345.6; 151 4325.67;"
formatSpec(x, fromTo = "str2mat")
```

**Description**

Print out some useful header information, such as instrumentation, ionization, range of mz and RT, and the number of MSn scans etc.

**Usage**

```r
getMetaInfo(filename)
```

**Arguments**

- **filename**

**Author(s)**

Mingshu Cao

**getMSnRaw**

**Get MSn raw data**

**Description**

Query MSn data by the attribute of 'msLevel' and get MSn raw data into a R list. Users may just use function "saveMSnRaw" to retrieve ion tree from a data file and avoid a direct Java function call.

**Usage**

```r
genMSnRaw(msdata, msLevel = 2)
```

**Arguments**

- **msdata**: msdata is a reference to a Java ArrayList. Obtained by calling getMSData Java function. msdata=java.call("XCMS", "Ljava/util/ArrayList;","getMSData", filename);
- **msLevel**: msLevel in integer
hasMS2

Value

- **premz**: ancestral precursor ions
- **rt**: retention time
- **msn.sp**: a list of spectrum (m/z, intensity)

Note

MS1 data could be queried by msLevel=1. The return type is still a list but different components (rt, tic, sp).

Author(s)

Mingshu Cao

---

hasMS2  
*Check which samples have MS2 spectra generated*

Description

check whether MS2 data are available for the ion/peak specified by mzRange and rtRange among samples.

Usage

```r
hasMS2(MS2RAW, mzRange = c(1854, 1854.5), rtRange = c(280, 400))
```

Arguments

- **MS2RAW**: MS2 raw data in R binary file, see `saveMSnRaw`
- **mzRange**: m/z range
- **rtRange**: rt range

Value

return sample index

Author(s)

Mingshu Cao
iontree-class

Class "iontree"

Description
iontree representation in S4 class

Objects from the Class
Objects can be created by calls of the form new("iontree", ...).

Slots
mz: Object of class "numeric" peak or ion m/z
rt: Object of class "numeric" peak or ion RT
MS2: Object of class "matrix" ms2 spectrum
MS3: Object of class "list" ms3 spectrum/spectra

Methods
plot signature(x = "iontree"): ...
show signature(object = "iontree"): ...

Note
To be extended to MSn where n>3

Author(s)
Mingshu Cao

Examples
showClass("iontree")

metaDataImport

Data entry of meta information

Description
data entry of table "experiment". Such meta information may include brief description of sample origin, biological treatment, extraction method, chromatography, ionization, polarity, collision energy and those might affect comparative analysis of iontrees. R default data editor was used to help provide necessary information. SQLite database browsers are also freely available for different platforms.

Usage
metaDataImport(dbname = "mzDB.db")
mzImport

Arguments

dbname database name

Note

A known issue: a call to use data editor (fix) might cause access violation, that was occasionally observed.

Author(s)

Mingshu Cao

Examples

# to check information just loaded
# db=dbConnect(dbDriver("SQLite"), dbname="mzDB.db")
# dbListTables(db)
# q1=dbSendQuery(db, "SELECT * FROM experiment")
# fetch(q1, n=-1)
# dbClearResult(q1)
# dbDisconnect(db)

mzImport Data entry of iontree into mz table

Description

import iontree object into table "mz" in the database

Usage

mzImport(iontree, dbname = "mzDB.db", exp.id)

Arguments

iontree iontree as defined in this package
dbname database name
exp.id id in table of experiment

Note

see vignette for an example of batch loading

Author(s)

Mingshu Cao
Examples

# to check information just loaded
# db = dbConnect(dbDriver("SQLite"), dbname="mzDB.db")
# dbListTables(db)
# q1 = dbSendQuery(db, "SELECT mz, rt, ms2 FROM mz")
# fetch(q1, n=-1)
# dbClearResult(q1)
# dbDisconnect(db)

Description

plot iontree

Methods

signature(x = "iontree") plot spectral tree

Description

plot a mass spectrum

Usage

plotSpectrum(x, y=NULL, top = 20, type = "h", scale100 = FALSE,
             digit.label = 1, col = "black", pos = 0, main = "",
             clickAddLabels = FALSE, ...)

Arguments

x m/z
y intensity
top top intense m/z to be labelled
type plot type
scale100 in scale of 0-100 if true
digit.label m/z precision to be maintained
col m/z label color
pos m/z label position
main title
clickAddLabels click to add labels
... as in plot

Author(s)

Mingshu Cao
rs2iontree

Convert a data frame into iontree

Description
Convert resultset, a data frame retrieved from database into a list of iontree objects.

Usage
rs2iontree(rs)

Arguments
rs
resultset as a data frame

Value
a list of iontrees

Author(s)
Mingshu Cao

saveMSnRaw
Retrieve MS2 and MS3 data and save as R binary in the current folder

Description
save MS2 and MS3 data for later processing such as ion tree construction. R binary files "MS2RAW.Rdata" and "MS3RAW.Rdata" may be found in the current folder, which can be reloaded.

Usage
saveMSnRaw(dataFolder = "D:/Data/Raw")

Arguments
dataFolder
current data folder

Author(s)
Mingshu Cao

Examples
#saveMSnRaw("D:/Data/Raw")
#load("D:/Data/Raw/MS2RAW.Rdata")
#ls()
searchMS2  

Search MS2 spectrum from sqlite database

Description

search MS2 spectra from the database. Ranking is based on the distance metric that defined by Cao et al. 2008, cosine and Tanimoto similarity.

Usage

searchMS2(querySpec, premz, dbname = "mzDB.db", scoreFun = "distMS2", output.record = 5, plot.top = TRUE)

Arguments

querySpec  query spectrum in 2-col matrix
premz  precursor m/z that query spectrum derived from
dbname  database name
scoreFun  score function, 'distMS2','cos' or 'tanimoto'
output.record  the number of records shown in console
plot.top  plot query spectrum and the top-ranked spectrum

Value

return top records

Author(s)

Mingshu Cao

show-methods  

Show methods for class iontree

Description

show methods for class iontree

Methods

signature(object = "iontree") show iontree object
topIons

Retain spectrum with the most intense ions

Description
Retain the most intense ions in a spectrum and return sorted spectrum in 2-col matrix

Usage
topIons(mz, intensity, top)

Arguments
- mz: m/z
- intensity: intensity
- top: the number of most intense m/z to be maintained

Value
return a 2-col matrix

Author(s)
Mingshu Cao

Examples
# Just sort:
# topIons(mz, intensity, top=length(mz))
Index

*Topic **IO**
  - createDB, 3
  - saveMSnRaw, 10
*Topic **aplot**
  - plotSpectrum, 9
*Topic **classes**
  - iontree-class, 7
*Topic **cluster**
  - distMS2, 4
*Topic **manip**
  - searchMS2, 11
*Topic **methods**
  - plot-methods, 9
  - show-methods, 11
*Topic **package**
  - iontree-package, 2
*Topic **utilities**
  - buildIonTree, 2
  - createDB, 3
  - distMS2, 4
  - formatSpec, 4
  - getMetaInfo, 5
  - getMSnRaw, 5
  - hasMS2, 6
  - metaDataImport, 7
  - mzImport, 8
  - rs2iontree, 10
  - saveMSnRaw, 10
  - topIons, 12

buildIonTree, 2
createDB, 3
distMS2, 4
formatSpec, 4
getMetaInfo, 5
getMSnRaw, 5
hasMS2, 6
iontree (iontree-package), 2
iontree-class, 7

iontree-package, 2
metaDataImport, 7
mzImport, 8
plot, iontree-method (iontree-class), 7
plot-methods, 9
plotSpectrum, 9
rs2iontree, 10
saveMSnRaw, 10
searchMS2, 11
show, iontree-method (iontree-class), 7
show-methods, 11
topIons, 12