Package: LOBSTAHS (via r-universe)

November 25, 2026

Title Lipid and Oxylipin Biomarker Screening through Adduct Hierarchy Sequences

Version 1.40.0

Date 2021-08-30

Depends R (>= 3.4), xcms, CAMERA, methods

Imports utils

Suggests PtH2O2lipids, knitr, rmarkdown

Description LOBSTAHS is a multifunction package for screening, annotation, and putative identification of mass spectral features in large, HPLC-MS lipid datasets. In silico data for a wide range of lipids, oxidized lipids, and oxylipins can be generated from user-supplied structural criteria with a database generation function. LOBSTAHS then applies these databases to assign putative compound identities to features in any high-mass accuracy dataset that has been processed using xcms and CAMERA. Users can then apply a series of orthogonal screening criteria based on adduct ion formation patterns, chromatographic retention time, and other properties, to evaluate and assign confidence scores to this list of preliminary assignments. During the screening routine, LOBSTAHS rejects assignments that do not meet the specified criteria, identifies potential isomers and isobars, and assigns a variety of annotation codes to assist the user in evaluating the accuracy of each assignment.

License GPL (>= 3) + file LICENSE

URL http://bioconductor.org/packages/LOBSTAHS

BugReports https://github.com/vanmooylipidomics/LOBSTAHS/issues/new

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Metabolomics, Lipidomics, DataImport

NeedsCompilation no

Author James Collins [aut, cre], Helen Fredricks [aut], Bethanie Edwards [aut], Henry Holm [aut], Benjamin Van Mooy [aut], Daniel Lowenstein [aut]

Maintainer Henry Holm <hholm@whoi.edu>, Daniel Lowenstein <dlowenstein@whoi.edu>, James Collins <james.r.collins@aya.yale.edu>

Config/pak/sysreqs
cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev libnetcdf-dev libssl-dev zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:37:39 UTC

RemoteUrl https://github.com/bioc/LOBSTAHS

RemoteRef RELEASE_3_24

RemoteSha e138113c0369d956f4567675ed92c713cac41f1e

Contents

doLOBscreen 2
exportDBtoCSV 5
extractLOBdbasedata 6
generateLOBdbase 7
getLOBpeaklist 11
loadLOBdbase 13
LOBdbase 15
LOBdbase-class 17
LOBdefaults 19
LOBSet 21
LOBSet-class 23

Index **25**

`doLOBscreen` *Screen, annotate, and identify compounds in an xsAnnotate object*

Description

Primary function of the `LOBSTAHS` package. Screen, annotate, and assign compound identities to peak groups in a CAMERA `xsAnnotate` object containing HPLC-ESI-MS lipid data. Identify and annotate possible isomers and isobars.

Usage

```
doLOBscreen(xsA, polarity = NULL, database = NULL, remove.iso = TRUE,
            rt.restrict =  TRUE, rt.windows = NULL, exclude.oddFA = TRUE,
           exclude.oxyrule =TRUE, match.ppm = NULL, retain.unidentified = TRUE)
```

Arguments

xsA	A CAMERA xsAnnotate-class object with identified pseudospectra. Must contain isotope data from findIsotopes if user elects remove.iso = TRUE).
polarity	Specify polarity mode of data in xsA: "positive" or "negative." doLOBscreen will attempt to detect the polarity mode if nothing is given.
database	Specify the LOBdbase from which compound identities and adduct ion hierarchy data are to be applied. If nothing is specified, doLOBscreen will use the LOBSTAHS default database (default.LOBdbase) for the appropriate ionization mode. The database generation function generateLOBdbase can be used to create a LOBdbase from structural property ranges specified by the user in a series of input tables. loadLOBdbase can be used to import and reconstruct a "LOBdbase" object from a previously generated database that was saved as a .csv file.
remove.iso	Should secondary isotope peaks be removed? (If TRUE, xsA must contain isotope data obtained using findIsotopes.)
rt.restrict	Should lipid class retention time screening criteria be applied to putative compound assignments?
rt.windows	File path to a .csv file containing retention time "window" data to be used for screening by lipid class, if rt.restrict = TRUE. If nothing is specified, doLOBscreen will use the package default windows (default.rt.windows), which are specific to the HPLC-MS method currently used in the Van Mooy Lab at Woods Hole Oceanographic Institution. Because the user is advised to provide retention time data specific to his/her HPLC-MS method, failure to specify a value for rt.windows will result in a warning. A Microsoft Excel spreadsheet template included with the package at Resources/library/LOBSTAHS/doc/xlsx/LOBSTAHS_rt.windows.xlsx can be used to generate a .csv file of retention time data in a format appropriate for rt.windows. Alternatively, the spreadsheet file may be downloaded from the package GitHub repository.
exclude.oddFA	Should compound assignments with an odd total number of acyl carbon atoms be excluded? (Applies only to assignments where the parent lipid class is TAG, IP-DAG, PUA, FFA or IP-MAG.) Useful if data are (or are believed to be) of exclusively eukaryotic origin.
match.ppm	*m/z* tolerance (in ppm) used for matching observed data against the calculated *m/z*'s in the database
exclude.oxyrule	Should compound assignments include physically unlikey fatty acid molecules in which the total number of double bonds is less than the oxidation state of the molecule? Useful if such rare molecules are likely to be found in sample. Exclusion applies only to assignments where the parent lipid class is TAG, IP-DAG, PUA, FFA or IP-MAG.
retain.unidentified	Should data for unidentified or discarded features be retained and appended to user's results? Useful if untargeted follow-on data analysis is anticipated, or when user wants to retain data for *all* features present in the original dataset, not just those for which a LOBSTAHS identity was found.

Details

doLOBscreen draws compound identities from a LOBdbase database. The function applies various retention time and adduct ion hierarchy screening criteria to winnow the list of putative compound assignments. It returns a table of annotated peak data with compound assignments and various annotation codes to assist the user in interpretation and follow-on data analysis.

Value

A "LOBSet-class" object.

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

The LOBSTAHS package is presented in:

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162, doi:10.1021/acs.analchem.6b01260.

Data for lipid classes BLL, PDPT, vGSL, sGSL, hGSL, hapGSL, and hapCER are as described in:

Hunter J. E., M. J. Frada, H. F. Fredricks, A. Vardi, and B. A. S. Van Mooy. 2015. Targeted and untargeted lipidomics of *Emiliania huxleyi* viral infection and life cycle phases highlights molecular biomarkers of infection, susceptibility, and ploidy. *Frontiers in Marine Science* 2:81, doi:10.3389/fmars.2015.00081

Fulton, J. M., H. F. Fredricks, K. D. Bidle, A. Vardi, B. J. Kendrick, G. R. DiTullio, and B. A. S. Van Mooy. 2014. Novel molecular determinants of viral susceptibility and resistance in the lipidome of *Emiliania huxleyi*, *Environmental Microbiology* 16(4):1137-1149, doi:10.1111/1462-2920.12358.

See Also

LOBSet, LOBdbase, loadLOBdbase, getLOBpeaklist, generateLOBdbase, default.LOBdbase, default.rt.windows, xcmsSet, xsAnnotate, findIsotopes

Examples

```
## screen & annotate xsAnnotate object from the PtH2O2lipids dataset using all
## screening options, while retaining unidentified/discarded features
library(PtH2O2lipids)


myPtH202LOBSet = doLOBscreen(ptH2O2lipids$xsAnnotate, polarity = "positive",
                             database = NULL, remove.iso = TRUE,
                             rt.restrict =  TRUE, rt.windows = NULL,
                             exclude.oddFA = TRUE, match.ppm = 2.5,
                             exclude.oxyrule = TRUE, retain.unidentified = TRUE)

## show some diagnostics
LOBscreen_diagnostics(myPtH202LOBSet) # screening diagnostics
```

```
LOBisoID_diagnostics(myPtH202LOBSet) # isomer identification data
```

`exportDBtoCSV` *Export a LOBSTAHS database (LOBdbase) object to a text file*

Description

Collects all elements of a `LOBdbase-class` object into a single table, then writes the table to a .csv file.

Usage

```
exportDBtoCSV(LOBdbase)
```

Arguments

`LOBdbase`	A LOBSTAHS `"LOBdbase"` object.

Details

`exportDBtoCSV` extracts data from all slots in a given `"LOBdbase"` object and flows it into a data frame. The data frame is then written to a .csv file with a series of plain-language column headers. The file name includes the database polarity mode and a timestamp (in user's local time) indicating the time and date of export.

Value

A .csv file containing data from a `LOBdbase-class` object.

Author(s)

James Collins, `<james.r.collins@aya.yale.edu>`

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

`LOBdbase`, `LOBdbase`, `generateLOBdbase`

Examples

```
## export default positive mode database to a text file

data("default.LOBdbase")

exportDBtoCSV(default.LOBdbase$positive)

## file is written to user's working directory, with filename containing
## polarity and timestamp of generation
```

`extractLOBdbasedata` *Extract compound data from a LOBdbase*

Description

Extracts data from a LOBSTAHS `LOBdbase-class` object for a given molecular species.

Usage

```
extractLOBdbasedata(frag_ID, database)
```

Arguments

`frag_ID`	The LOBdbase fragment ID(s) corresponding to the species for which data should be extracted.
`database`	The `LOBdbase` from which data are to be extracted.

Value

A data frame containing the extracted data, with field names corresponding to the slots in `database`.

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

`LOBdbase`, `LOBdbase loadLOBdbase`, `generateLOBdbase`, `default.LOBdbase`

Examples

```
## extract data from the default database for the [M+H]+ adduct of
## dinitrophenyl-phosphatidylethanolamine (DNPPE), a synthetic lipid used as an
## internal standard

data(default.LOBdbase)

DNPPE.data = extractLOBdbasedata(frag_ID = 19801,
                                 database = default.LOBdbase$positive)
```

`generateLOBdbase` *Conduct* in silico *simulation and generate lipid-oxylipin database*

Description

Applies an *in silico* simulation to generate data by ionization mode (polarity) for a wide range of lipids, oxidized lipids, and oxylipins. User-supplied structural criteria and empirically-determined adduct ion abundance rankings for the major lipid classes are used to create entries for a range of lipid moieties. The database(s) can then be used in `doLOBscreen` to assign compound identities to grouped peakdata.

Usage

```
generateLOBdbase(polarity = c("positive","negative"), gen.csv = FALSE,
                 component.defs = NULL, AIH.defs = NULL, acyl.ranges = NULL,
                 oxy.ranges = NULL)
```

Arguments

`polarity`	Ionization mode for which database is to be generated.
`gen.csv`	Should results also be written to a .csv file?
`component.defs`	File path to a .csv file containing elemental composition definitions for the various chemical components needed by `generateLOBdbase`. If nothing is specified, `generateLOBdbase` will use the default composition table (`default.componentCompTable`). The default table includes definitions for the masses of a wide range of adducts, photosynthetic pigments, and structural backbones of some major lipid classes. A Microsoft Excel spreadsheet template included with the package at Resources/library/LOBSTAHS/doc/ can be used to generate a .custom csv file with elemental composition definitions in a format appropriate for `generateLOBdbase`. Alternatively, the spreadsheet may be downloaded from the package GitHub repository. Brief instructions for customization of the table are given in this help document; full instructions, including details on specification of the necessary base fragment, are contained in the package vignette. For each lipid class or compound specified in the component definitions table, the field `DB_gen_compound_type` must contain one of five values: "DB_acyl_iteration,"

"DB_unique_species," "basic_component," "adduct_pos," or "adduct_neg." The last three compound types are reserved for definition of basic components such as acteonitrile or acetate and for definition of adduct ion types; new entries of these types should only be created in the compound table when a new adduct or basic component must be specified. The first two compound types are used to define the way `generateLOBdbase` creates its databases. There are essentially two ways `generateLOBdbase` creates database entries in LOBSTAHS.

For compounds of `DB_gen_compound_type` = "DB_unique_species" (the simpler of the two cases), database entries will be created only for adduct ions of the single compound specified. The latter type should be used for pigments and other lipids that do not have acyl groups, or when the user does not wish to consider any possible variation in acyl properties. In this case, the exact mass of the complete (neutral) molecule should be specified in the component definitions (i.e., component composition) table.

Alternatively, for compounds of `DB_gen_compound_type` = "DB_acyl_iteration", `generateLOBdbase` will create database entries for adduct ions of multiple molecular species within the lipid class based on the ranges of acyl properties and oxidation states given for the class in `acyl.ranges` and `oxy.ranges` (see below). In this case, the compound table should be used to define the exact mass of a "base fragment" for the lipid class. Using this "base fragment" as a starting point, `generateLOBdbase` creates multiple entries for molecules in the lipid class by iterative addition of various combinations of fatty acids. In the case of IP-DAG and IP-MAG, the base fragment includes the entire polar headgroup, the glycerol backbone, and both carboxylic oxygen atoms in the fatty acid(s). In the case of TAG, the base fragment is defined as the glycerol backbone plus the carboxylic oxygen atoms on each of the three fatty acids. The base fragments for any new lipid classes for which the user desires evaluation of a range of acyl properties should be similarly defined.

Note that regardless of the `DB_gen_compound_type`, an adduct hierarchy **must** be specified in the adduct ion hierarchy matrix (see below) for each compound or compound class specified in the `Adduct_hierarchy_lookup_class` field of the component definitions table.

`AIH.defs` File path to a .csv file containing empirical adduct ion hierarchy (AIH) data for various pigments, lipids, and lipid classes. If nothing is specified, `generateLOBdbase` will use the default AIH data (`default.adductHierarchies`). Each compound or compound class for which there is an entry in the AIH definitions table should have at least one corresponding entry in the `Adduct_hierarchy_lookup_class` field of the component definitions table (default, `default.componentCompTable`).

A Microsoft Excel spreadsheet template included with the package at Resources/library/LOBSTAHS/doc/ can be used to generate a .csv file with additional (or alternative) adduct hierarchy data in a format appropriate for `generateLOBdbase`. Alternatively, the spreadsheet may be downloaded from the package GitHub repository.

`acyl.ranges` File path to a .csv file containing ranges of values for the total number of acyl (i.e., fatty acid) carbon atoms to be considered during the *in silico* simulation of any lipid classses for which `DB_gen_compound_type` is specified as "DB_acyl_iteration" in the component definitions table, above. These include intact polar diacylglycerols (IP-DAG), triacylglycerols (TAG), polyunsaturated

aldehydes (PUAs), and free fatty acids (FFA). If nothing is specified, `generateLOBdbase` will use the default acyl carbon atom range data in (`default.acylRanges`).

A Microsoft Excel spreadsheet template included with the package at Resources/library/LOBSTAHS/doc/ can be used to generate a .csv file with custom acyl carbon range data in a format appropriate for `generateLOBdbase`. Alternatively, the spreadsheet may be downloaded from the package GitHub repository.

`oxy.ranges` File path to a .csv file containing ranges of values for the number of additional oxygen atoms to be considered during the *in silico* simulation of any lipid classses for which `DB_gen_compound_type` is specified as "DB_acyl_iteration" in the component definitions table, above. If nothing is specified, `generateLOBdbase` will use the default oxidation state ranges in (`default.oxyRanges`).

A Microsoft Excel spreadsheet template included with the package at Resources/library/LOBSTAHS/doc/ can be used to generate a .csv file with custom oxidation state ranges in a format appropriate for `generateLOBdbase`. Alternatively, the spreadsheet may be downloaded from the package GitHub repository. By default, `generateLOBdbase` considers 0-4 additional oxygen atoms on each chemically possible IP-DAG, TAG, PUA, and FFA.

Details

Using the default structural property inputs described here, `generateLOBdbase` can produce databases with entries for a wide range of intact polar diacylglycerols (IP-DAG), triacylglycerols (TAG), polyunsaturated aldehydes (PUAs), free fatty acids (FFA), and common photosynthetic pigments. The default databases (as of January 2017) contain data on 18,067 and 15,404 unique compounds that can be identifed in positive and negative ion mode spectra, respectively.

Note that the default databases have been pre-generated (see `default.LOBdbase`) and it is therefore unnecessary to call `generateLOBdbase` with the default parameters.

Value

A "`LOBdbase-class`" object with the structure:

`frag_ID`**:** Object of class "`integer`", a unique identifier for this molecular species

`mz`**:** Object of class "`numeric`", the calculated *m/z* of this species

`exact_parent_neutral_mass`**:** Object of class "`numeric`", the calculated (monoisotopic) exact mass of the parent compound of this species

`lipid_class`**:** Object of class "`factor`", the parent lipid class of this species

`species`**:** Object of class "`character`", the lipid subclass

`adduct`**:** Object of class "`factor`", the adduct ion represented by this entry

`adduct_rank`**:** Object of class "`integer`", the relative abundance ranking of this adduct relative to the other adducts of the same parent compound

`FA_total_no_C`**:** Object of class "`integer`", total number of acyl (fatty acid) carbon atoms in the parent compound; `NA` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`FA_total_no_DB`**:** Object of class "`integer`", total number of acyl (fatty acid) carbon-carbon double bonds in the parent compound; `NA` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`degree_oxidation`**:** Object of class "`integer`", number of additional oxygen atoms present

`parent_elem_formula`: Object of class `"character"`, elemental formula of the parent compound

`parent_compound_name`: Object of class `"character"`, name of the parent compound; see the reference for this entry for the naming convention applied to compounds other than pigments

`polarity`: Object of class `"factor"`, ionization mode of data in the database

`num_entries`: Object of class `"integer"`, number of total entries (adducts) in the database

`num_compounds`: Object of class `"integer"`, number of parent compounds represented in the database (should be < `num_entries`)

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

The LOBSTAHS package is presented in:

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162, doi:10.1021/acs.analchem.6b01260.

Data for lipid classes BLL, PDPT, vGSL, sGSL, hGSL, hapGSL, and hapCER are as described in:

Hunter J. E., M. J. Frada, H. F. Fredricks, A. Vardi, and B. A. S. Van Mooy. 2015. Targeted and untargeted lipidomics of *Emiliania huxleyi* viral infection and life cycle phases highlights molecular biomarkers of infection, susceptibility, and ploidy. *Frontiers in Marine Science* 2:81, doi:10.3389/fmars.2015.00081

Fulton, J. M., H. F. Fredricks, K. D. Bidle, A. Vardi, B. J. Kendrick, G. R. DiTullio, and B. A. S. Van Mooy. 2014. Novel molecular determinants of viral susceptibility and resistance in the lipidome of *Emiliania huxleyi*, *Environmental Microbiology* 16(4):1137-1149, doi:10.1111/1462-2920.12358.

See Also

`LOBdbase`, `LOBdbase`, `loadLOBdbase`, `doLOBscreen`, `default.LOBdbase`, `default.componentCompTable`, `default.adductHierarchies`, `default.acylRanges`, `default.oxyRanges`

Examples

```
## generate the default positive ionization mode database


LOBdbase.pos = generateLOBdbase(polarity = "positive", gen.csv = FALSE,
                                component.defs = NULL, AIH.defs = NULL,
                                acyl.ranges = NULL, oxy.ranges = NULL)
```

getLOBpeaklist *Export screened LOBSTAHS peaklist with compound assignments*

Description

Extracts screened peak data, compound assignments, annotation codes, and (optionally) isomer data from a LOBSet-class object. Returns a table containing the requested data, with option to export to a .csv file.

Usage

```
getLOBpeaklist(LOBSet, include.iso = TRUE, include.unidentified = TRUE,
               gen.csv = FALSE)
```

Arguments

LOBSet	A LOBSTAHS "LOBSet" object.
include.iso	Specify whether isomer and isobar identification data should be included in peaklist.
include.unidentified	If data for unidentified or discarded features is present in the LOBSet, should these data be exported with the peaklist? A value of FALSE will yield a peaklist that includes data only for those features to which one or more LOBSTAHS compound identities have been assigned.
gen.csv	Should a .csv file be generated in addition to the data frame that is returned?

Details

getLOBpeaklist extracts data from all slots in a given "LOBSet" object and flows it into a data frame. Annotation codes indicating compliance with the adduct ion hierarchy screening criteria are appended to the *m/z*, retention time, and peak area data for each xcms peakgroup remaining in the final dataset.

If include.iso = TRUE, three additional columns containing the match_IDs of the possible isomers and isobars for each peakgroup are also appended. getLOBpeaklist does not export any of the diagnostic data in the LOBSet LOBscreen_diagnostics or LOBisoID_diagnostics slots.

When the LOBSet includes peak data for features that were not identified during screening or were discarded (i.e., if the user specified retain.unidentified = TRUE when calling doLOBscreen), the user is given the option to export these data when using getLOBpeaklist. This is useful when untargeted follow-on data analysis is anticipated, or the user simply wants to export data for *all* features present in the original dataset, not just those for which a LOBSTAHS identity was found.

Value

A data frame with the following structure:

`match_ID`: Object of class `"integer"`, unique identifier for each assignment of a compound to a peakgroup (multiple `match_ID`s can exist for a peakgroup if the group was assigned multiple compound identities

`compound_name`: Object of class `"character"`, name of compound; see reference for naming convention applied to compounds other than pigments

`elem_formula`: Object of class `"character"`, empirical formula of compound

`LOBdbase_mz`: Object of class `"numeric"`, calculated *m/z* of the adduct for which data in this group are reported; obtained database

`peakgroup_mz`: Object of class `"numeric"`, mean observed *m/z* of the feature in this peakgroup across all samples in which it was identified

`LOBdbase_ppm_match`: Object of class `"numeric"`, ppm deviation between observed and calculated *m/z*

`peakgroup_rt`: Object of class `"numeric"`, mean observed retention time of the feature in this peakgroup across all samples in which it was identified

`peakgroup_mzmin`: Object of class `"numeric"`, minimum observed *m/z* of feature across samples

`peakgroup_mzmax`: Object of class `"numeric"`, maximum observed *m/z* of feature across samples

`peakgroup_rtmin`: Object of class `"numeric"`, minimum observed retention time of feature across samples

`peakgroup_rtmax`: Object of class `"numeric"`, maximum observed retention time of feature across samples

peak area data: Several objects of class `"numeric"`, containing integrated peak area data for this group by sample (one column for each sample in the dataset)

`xcms_peakgroup`: Object of class `"integer"`, the xcms `xcmsSet` peakgroup identifier

`CAMERA_pseudospectrum`: Object of class `"integer"`, the CAMERA `xsAnnotate` pseudospectrum identifier

`LOBdbase_frag_ID`: Object of class `"integer"`, the `LOBdbase` fragment ID corresponding to the adduct of this compound for which data are reported; this is the dominant adduct of the compound according to the adduct ion hierarchy rules for the parent lipid class

`LOBdbase_exact_parent_neutral_mass`: Object of class `"numeric"`, the calculated exact (monoisotopic) mass of the compound; from database

`lipid_class`: Object of class `"factor"`, parent lipid class of this compound

`species`: Object of class `"character"`, if a pigment or IP-DAG, the specific compound species

`major_adduct`: Object of class `"factor"`, adduct of the compound for which data in this entry is reported

`FA_total_no_C`: Object of class `"integer"`, total number of acyl (fatty acid) carbon atoms in this compound; `"NA"` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`FA_total_no_DB`: Object of class `"integer"`, total number of acyl (fatty acid) carbon-carbon double bonds in this compound; `"NA"` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`degree_oxidation`: Object of class `"integer"`, number of additional oxygen atoms present on this compound, compared with its unoxidized parent

`C1-C6b`: Several objects of class `"integer"`, containing binary indicators for each possible annotation code applied by `doLOBscreen`

casecodes: Object of class "character", character string containing list of all codes applied to this assignment

iso_C3r_match_ID: Object of class "character", character string of integer containing the match_IDs of all possible regioisomers of this compound (if include.iso = TRUE)

iso_C3f_match_ID: Object of class "character", character string of integer containing the match_IDs of all possible functional structural isomers of this compound (if include.iso = TRUE)

iso_C3c_match_ID: Object of class "character", character string of integer containing the match_IDs of all possible isobars of this compound (if include.iso = TRUE)

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

LOBSet, LOBSet, doLOBscreen

Examples

```
## export peaklist, with isomer data, but without data for unidentified features
library(PtH2O2lipids)

PtH2O2.peakdata = getLOBpeaklist(ptH2O2lipids$LOBSet, include.iso = TRUE,
                                 include.unidentified = FALSE, gen.csv = FALSE)
```

---

loadLOBdbase *Import and reconstruct LOBdbase from a text file*

Description

Reconstruct a "LOBdbase" object from properly formatted data in a .csv file. More robust than the generic constructor function LOBdbase.

Usage

```
loadLOBdbase(file, polarity, num_compounds = NULL)
```

Arguments

file	Path to a .csv file containing the database to be imported. The file format should be consistent with the .csv output that is obtained using gen.csv = TRUE in generateLOBdbase. Some leeway is allowed for variation in capitalization and punctuation of column headers.
polarity	Specify polarity (ionization mode) of the database being imported ("positive" or "negative"). loadLOBdbase will attempt to detect the polarity mode based on data in the "adduct" column if nothing is given.
num_compounds	If known, the number of parent compounds represented in the database being imported. Can be unspecified.

Details

loadLOBdbase is a glorified implementation of read.table that attempts to determine whether the format of the data in file is consistent with that of a saved "LOBdbase" object. Some leeway is allowed for variation in punctuation and spelling of column headers in the source file. Special attention should be paid to how adduct ions are specified (e.g., "[M+H]+", "[M+NH4]+", or "[M+Cl]-"). Most users will find this function more useful than the generic constructor function LOBdbase.

Value

A "LOBdbase-class" object.

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

LOBdbase, LOBdbase, doLOBscreen, generateLOBdbase, default.LOBdbase

Examples

```
## save the default negative mode database as a .csv file

data(default.LOBdbase)

neg.DB = default.LOBdbase$negative

fname = paste0("LOBSTAHS_lipid-oxy_DB_",
               strtrim(as.character(polarity(neg.DB)),3),".csv")

exportmat = data.frame(frag_ID(neg.DB),
                       mz(neg.DB),
```

```
                          exact_parent_neutral_mass(neg.DB),
                          as.character(lipid_class(neg.DB)),
                          as.character(species(neg.DB)),
                          as.character(adduct(neg.DB)),
                          as.character(adduct_rank(neg.DB)),
                          FA_total_no_C(neg.DB),
                          FA_total_no_DB(neg.DB),
                          degree_oxidation(neg.DB),
                          parent_elem_formula(neg.DB),
                          parent_compound_name(neg.DB),
                          stringsAsFactors = FALSE)

colnames(exportmat) = c("frag_ID","mz","exact_parent_neutral_mass",
                        "lipid_class","species","adduct","adduct_rank",
                        "FA_total_no_C","FA_total_no_DB","degree_oxidation",
                        "parent_elem_formula","parent_compound_name")

write.csv(exportmat, fname)

## reimport it

neg.DB.reimported = loadLOBdbase("LOBSTAHS_lipid-oxy_DB_neg.csv",
                                 polarity = "negative",
                                 num_compounds = NULL)
```

LOBdbase *LOBdbase constructor for manual creation or reconstruction of a LOBdbase object*

Description

Constructor function for manual assembly or reconstruction of a LOBdbase object to be used in screening data with the LOBSTAHS function doLOBscreen.

Usage

```
LOBdbase(frag_ID = NULL, mz = NULL, exact_parent_neutral_mass = NULL,
        lipid_class = NULL, species = NULL, adduct = NULL, adduct_rank = NULL,
        FA_total_no_C = NULL, FA_total_no_DB = NULL, degree_oxidation = NULL,
        parent_elem_formula = NULL, parent_compound_name = NULL,
        polarity = NULL, num_entries = NULL, num_compounds = NULL)
```

Arguments

frag_ID	An object of class "integer"; vector of unique identifiers for the molecular species in the database.
mz	An object of class "numeric"; calculated *m/z* of each species for which an entry exists.

`exact_parent_neutral_mass`
: An object of class `"numeric"`; calculated (monoisotopic) exact masses of the parent compound of each species.

`lipid_class`
: An object of class `"factor"`; parent lipid classes of each species

`species`
: An object of class `"character"`; the lipid subclasses of each species

`adduct`
: An object of class `"factor"`; the adduct ions represented by each entry

`adduct_rank`
: An object of class `"integer"`; the relative abundance rankings of each adduct relative to the other adducts of the same parent compound

`FA_total_no_C`
: An object of class `"integer"`; the total number of acyl (fatty acid) carbon atoms in the parent compound of each entry; values should be `NA` where `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`FA_total_no_DB`
: An object of class `"integer"`; the total number of acyl (fatty acid) carbon-carbon double bonds in the parent compound of each entry; values should be `NA` where `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`degree_oxidation`
: An object of class `"integer"`; the number of additional oxygen atoms present on each species

`parent_elem_formula`
: An object of class `"character"`; the elemental formulae of the parent compound of each species

`parent_compound_name`
: An object of class `"character"`; names of the parent compound of each species; see the reference for this entry for snaming conventions that should be applied for pigments and compounds other than pigments

`polarity`
: An object of class `"factor"` with length = 1; the ionization mode of data in the database to be constructed

`num_entries`
: An object of class `"integer"` with length = 1; the number of total entries (adducts) in the database to be constructed. If specified, the value of `num_entries` should be equal to the length of any arguments that contain database data

`num_compounds`
: An object of class `"integer"` with length = 1; the number of parent compounds represented in the database to be constructed. If specified, the value of `num_compounds` should be < the value specifed for `num_entries`.

Details

Typically, a `LOBdbase` will be created using the in silico simulation function `generateLOBdbase`. Formatted database entries (such as from an external .csv file) can be loaded using `loadLOBdbase`. The rudimentary `LOBdbase` constructor function is therefore provided only for manual object creation; it will not be needed by most users. All arguments except for those containing metadata (i.e., `polarity`, `num_entries`, and `num_compounds`) should be of the same length.

Value

A `"LOBdbase-class"` object.

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

The LOBSTAHS package is presented in:

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162, doi:10.1021/acs.analchem.6b01260.

Data for lipid classes BLL, PDPT, vGSL, sGSL, hGSL, hapGSL, and hapCER are as described in:

Hunter J. E., M. J. Frada, H. F. Fredricks, A. Vardi, and B. A. S. Van Mooy. 2015. Targeted and untargeted lipidomics of *Emiliania huxleyi* viral infection and life cycle phases highlights molecular biomarkers of infection, susceptibility, and ploidy. *Frontiers in Marine Science* 2:81, doi:10.3389/fmars.2015.00081

Fulton, J. M., H. F. Fredricks, K. D. Bidle, A. Vardi, B. J. Kendrick, G. R. DiTullio, and B. A. S. Van Mooy. 2014. Novel molecular determinants of viral susceptibility and resistance in the lipidome of *Emiliania huxleyi*, *Environmental Microbiology* 16(4):1137-1149, doi:10.1111/1462-2920.12358.

See Also

generateLOBdbase, loadLOBdbase, doLOBscreen, LOBdbase

Examples

```
## create an empty LOBdbase

myLOBdbase = LOBdbase(frag_ID = NULL, mz = NULL,
                      exact_parent_neutral_mass = NULL, lipid_class = NULL,
                      species = NULL, adduct = NULL, adduct_rank = NULL,
                      FA_total_no_C = NULL, FA_total_no_DB = NULL,
                      degree_oxidation = NULL, parent_elem_formula = NULL,
                      parent_compound_name = NULL, polarity = NULL,
                      num_entries = NULL, num_compounds = NULL)
```

LOBdbase-class *Class LOBdbase: A class for LOBSTAHS lipid-oxylipin databases*

Description

A class for LOBSTAHS databases that contain a combination of *in silico* and empirical data for a wide range of lipids, oxidized lipids, and oxylipins.

Objects from the class

Objects can be created using the simulation function generateLOBdbase (preferred; satisfies the needs of most users), imported from a .csv file of proper format using loadLOBdbase, or created using the rudimentary constructor LOBdbase (least preferred). A 'LOBdbase' can also be created by calls of the form new("LOBdbase", ...).

Slots

`frag_ID`: Object of class `"integer"`, a unique identifier for this molecular species

`mz`: Object of class `"numeric"`, the calculated *m/z* of this species

`exact_parent_neutral_mass`: Object of class `"numeric"`, the calculated (monoisotopic) exact mass of the parent compound of this species

`lipid_class`: Object of class `"factor"`, the parent lipid class of this species

`species`: Object of class `"character"`, the lipid subclass

`adduct`: Object of class `"factor"`, the adduct ion represented by this entry

`adduct_rank`: Object of class `"integer"`, the relative abundance ranking of this adduct relative to the other adducts of the same parent compound

`FA_total_no_C`: Object of class `"integer"`, total number of acyl (fatty acid) carbon atoms in the parent compound; `NA` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`FA_total_no_DB`: Object of class `"integer"`, total number of acyl (fatty acid) carbon-carbon double bonds in the parent compound; `NA` if `lipid_class` is not TAG, IP-DAG, PUA, or FFA

`degree_oxidation`: Object of class `"integer"`, number of additional oxygen atoms present

`parent_elem_formula`: Object of class `"character"`, elemental formula of the parent compound

`parent_compound_name`: Object of class `"character"`, name of the parent compound; see the reference for this entry for the naming convention applied to compounds other than pigments

`polarity`: Object of class `"factor"`, ionization mode of data in the database

`num_entries`: Object of class `"integer"`, number of total entries (adducts) in the database

`num_compounds`: Object of class `"integer"`, number of parent compounds represented in the database (should be < `num_entries`)

Methods

show `signature(object = "LOBdbase")`: ...

polarity `signature(object = "LOBdbase")`: get `polarity` slot

num_compounds `signature(object = "LOBdbase")`: get `num_compounds` slot

num_entries `signature(object = "LOBdbase")`: get `num_entries` slot

frag_ID `signature(object = "LOBdbase")`: get `frag_ID` slot

exact_parent_neutral_mass `signature(object = "LOBdbase")`: get `exact_parent_neutral_mass` slot

lipid_class `signature(object = "LOBdbase")`: get `lipid_class` slot

species `signature(object = "LOBdbase")`: get `species` slot

adduct `signature(object = "LOBdbase")`: get `adduct` slot

adduct_rank `signature(object = "LOBdbase")`: get `adduct_rank` slot

FA_total_no_C `signature(object = "LOBdbase")`: get `FA_total_no_C` slot

FA_total_no_DB `signature(object = "LOBdbase")`: get `FA_total_no_DB` slot

degree_oxidation `signature(object = "LOBdbase")`: get `degree_oxidation` slot

parent_elem_formula `signature(object = "LOBdbase")`: get `parent_elem_formula` slot

parent_compound_name `signature(object = "LOBdbase")`: get `parent_compound_name` slot

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

generateLOBdbase, loadLOBdbase, doLOBscreen, LOBdbase

Examples

```
## return object information

showClass("LOBdbase")
```

LOBdefaults *LOBSTAHS default databases and database generation parameters*

Description

Default databases (in `default.LOBdbase`), and the .RData files containing default ranges of structural properties used by generateLOBdbase to generate these databases.

Note that the format of these files has changed in LOBSTAHS version 1.1.2 and greater.

Also described is `default.rt.windows`, which contains the default retention time windows (by lipid class) used by doLOBscreen when `rt.restrict = TRUE`.

Usage

```
data(default.LOBdbase)
data(default.adductHierarchies)
data(default.acylRanges)
data(default.oxyRanges)
data(default.componentCompTable)
data(default.rt.windows)
```

Format

`default.LOBdbase` A list of two "LOBdbase" objects, which are the default LOBSTAHS databases for positive and negative ion mode species, respectively. These were generated using generateLOBdbase with the default values defined in `default.adductHierarchies`, `default.acylRanges`, `default.oxyRanges`, and `default.componentCompTable`.

`default.adductHierarchies` A data frame containing empirically-determined adduct ion hierarchy data, by lipid class.

`default.acylRanges` A data frame containing ranges of numbers of acyl carbon atoms for which *in silico* data are generated for each lipid class by `generateLOBdbase`.

`default.oxyRanges` A data frame containing ranges of additional oxygen atoms to be considered on species of each lipid class when databases are generated with `generateLOBdbase`.

`default.componentCompTable` A data frame that defines the elemental compositions of the various adducts, parent lipid "backbones," and pigments that are used by `generateLOBdbase`.

`default.rt.windows` A data frame containing the default retention time data for various lipids and parent lipid classes that are used by `doLOBscreen` when `rt.restrict = TRUE`. These retention time windows are specific to the HPLC-MS method currently used in the Van Mooy Lab at Woods Hole Oceanographic Institution, where LOBSTAHS was developed. As described in `doLOBscreen`, users outside the Van Mooy Lab should supply their own retention time data.

Details

Empirical determination of the retention time window data in `default.rt.windows` and adduct ion hierarchies in `default.adductHierarchies` are described in the references below.

The default ranges for the structural properties given in the other files were chosen to yield databases that encompass a broad variety of moieites across lipid types. Microsoft Excel spreadsheet templates are included with the package in Resources/library/LOBSTAHS/doc for users wishing to modify any of the default data inputs. Alternatively, the spreadsheet files may be downloaded from a directory embedded within the package. These templates can be used to generate .csv files in formats appropriate for `generateLOBdbase` and `doLOBscreen`.

Value

Various list and data.frame objects (as indicated above).

Source

`http://github.com/vanmooylipidomics/LOBSTAHS/`

References

The LOBSTAHS package is presented in:

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162, doi:10.1021/acs.analchem.6b01260.

Data for lipid classes BLL, PDPT, vGSL, sGSL, hGSL, hapGSL, and hapCER are as described in:

Hunter J. E., M. J. Frada, H. F. Fredricks, A. Vardi, and B. A. S. Van Mooy. 2015. Targeted and untargeted lipidomics of *Emiliania huxleyi* viral infection and life cycle phases highlights molecular biomarkers of infection, susceptibility, and ploidy. *Frontiers in Marine Science* 2:81, doi:10.3389/fmars.2015.00081

Fulton, J. M., H. F. Fredricks, K. D. Bidle, A. Vardi, B. J. Kendrick, G. R. DiTullio, and B. A. S. Van Mooy. 2014. Novel molecular determinants of viral susceptibility and resistance in the lipidome of *Emiliania huxleyi*, *Environmental Microbiology* 16(4):1137-1149, doi:10.1111/1462-2920.12358.

See Also

doLOBscreen, generateLOBdbase, LOBdbase LOBdbase

LOBSet *LOBSet constructor for manual creation or reconstruction of a LOBSet object*

Description

Constructor function for manual creation or reconstruction of a LOBSet object for HPLC-MS peak data that have been screened using LOBSTAHS.

Usage

```
LOBSet(peakdata = NULL, iso_C3r = NULL, iso_C3f = NULL, iso_C3c = NULL,
       LOBscreen_diagnostics = NULL, LOBisoID_diagnostics = NULL,
       LOBscreen_settings = NULL, polarity = c("positive","negative"),
       sampnames = NULL)
```

Arguments

peakdata	An object of class "data.frame" containing peak data and LOBSTAHS annotation information by assignment. Column headings and data types should conform to those of the peakdata slot of a LOBSet-class object produced using the function doLOBscreen. Each row in the peakdata table represents one compound assignment made by LOBSTAHS. The format of the data frame occupying the peakdata slot in a LOBSet object can be obtained using the peakdata accessor for objects of LOBSet-class.
iso_C3r	An object of class "list", a list of the match_IDs of possible regioisomers of each compound for which there is a row in peakdata. Length of iso_C3r should equal the number of rows in the data frame given for peakdata.
iso_C3f	An object of class "list", a list of the match_IDs of possible functional structural isomers of each compound for which there is a row in peakdata. Length of iso_C3f should equal the number of rows in the data frame given for peakdata.
iso_C3c	An object of class "list", a list of the match_IDs of the isobars of each compound for which there is a row in peakdata. Length of iso_C3c should equal the number of rows in the data frame given for peakdata.
LOBscreen_diagnostics	An object of class "data.frame", containing diagnostic information recorded by the function doLOBscreen during creation of a LOBSet. The numbers of peaks, peakgroups, adducts, and unique parent compounds present the dataset after application of each LOBSTAHS screening criterion.
LOBisoID_diagnostics	An object of class "data.frame", containing isomer and isobar summary statistics. The numbers of peakgroups and parent compounds to which the various isomer identifications have been applied by doLOBscreen to a given LOBSet.

LOBscreen_settings
: An object of class "list"; the settings used in doLOBscreen to generate the LOBSet

polarity
: An object of class "factor", polarity of data in the LOBSet. Must be either "negative" or "positive."

sampnames
: An object of class "character"; the names of the samples from which the LOBSet was generated.

Details

Typically, a LOBSet will be created from a CAMERA xsAnnotate-class object using the LOBSTAHS function doLOBscreen. The LOBSet constructor function is therefore provided only for manual object creation; it will not be needed by most users.

Value

A "LOBSet-class" object.

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

LOBSet, doLOBscreen, xsAnnotate

Examples

```
## create an empty LOBSet for positive ion mode data

myLOBSet = LOBSet(peakdata = NULL, iso_C3r = NULL, iso_C3f = NULL,
                  iso_C3c = NULL, LOBscreen_diagnostics = NULL,
                  LOBisoID_diagnostics = NULL, LOBscreen_settings = NULL,
                  polarity = "positive", sampnames = NULL)
```

LOBSet-class *Class LOBSet: Peak data with annotations, isomers, and compound assignments*

Description

A class for HPLC-MS peak data that have been screened and annotated using LOBSTAHS function doLOBscreen.

Objects from the class

Objects can be created with the LOBSet constructor. A 'LOBSet' can also be created by calls of the form new("LOBSet", ...).

Slots

peakdata: Object of class "data.frame", containing peakdata by compound assignment

iso_C3r: Object of class "list", a list of the match_IDs of possible regioisomers of each compound

iso_C3f: Object of class "list", a list of the match_IDs of possible functional structural isomers of each compound

iso_C3c: Object of class "list", a list of the match_IDs of possible isobars of this compound

LOBscreen_diagnostics: Object of class "data.frame", numbers of peaks, peakgroups, adducts, and unique parent compounds present in the dataset after application of each screening criterion in doLOBscreen

LOBisoID_diagnostics: Object of class "data.frame", numbers of peakgroups and parent compounds to which the various isomer annotations have been assigned

LOBscreen_settings: Object of class "list", captures the settings used in doLOBscreen to generate the "LOBSet"

polarity: Object of class "factor", polarity of data in the "LOBSet"

sampnames: Object of class "character", the names of the samples from which the "LOBSet" was generated

Methods

show signature(object = "LOBSet"): ...

LOBisoID_diagnostics signature(object = "LOBSet"): get LOBisoID_diagnostics slot

LOBscreen_diagnostics signature(object = "LOBSet"): get LOBscreen_diagnostics slot

LOBscreen_settings signature(object = "LOBSet"): get LOBscreen_settings slot

sampnames signature(object = "LOBSet"): get sampnames slot

peakdata signature(object = "LOBSet"): get peakdata slot

polarity signature(object = "LOBSet"): get polarity slot

Author(s)

James Collins, <james.r.collins@aya.yale.edu>

References

Collins, J.R., B.R. Edwards, H.F. Fredricks, and B.A.S. Van Mooy. 2016. LOBSTAHS: An adduct-based lipidomics strategy for discovery and identification of oxidative stress biomarkers. *Analytical Chemistry* 88:7154-7162

See Also

doLOBscreen, getLOBpeaklist, LOBSet, xsAnnotate

Examples

```
## return object information

showClass("LOBSet")
```

Index

∗ **classes**
LOBdbase-class, 17
LOBSet-class, 23
∗ **datasets**
LOBdefaults, 19
∗ **methods**
doLOBscreen, 2
exportDBtoCSV, 5
extractLOBdbasedata, 6
generateLOBdbase, 7
getLOBpeaklist, 11
loadLOBdbase, 13
LOBdbase, 15
LOBSet, 21

adduct *(*LOBdbase-class*)*, 17
adduct,LOBdbase-method *(*LOBdbase-class*)*, 17
adduct<- *(*LOBdbase-class*)*, 17
adduct<-,LOBdbase-method *(*LOBdbase-class*)*, 17
adduct_rank *(*LOBdbase-class*)*, 17
adduct_rank,LOBdbase-method *(*LOBdbase-class*)*, 17
adduct_rank<- *(*LOBdbase-class*)*, 17
adduct_rank<-,LOBdbase-method *(*LOBdbase-class*)*, 17

default.acylRanges, *9, 10*
default.acylRanges *(*LOBdefaults*)*, 19
default.adductHierarchies, *8*, *10*
default.adductHierarchies *(*LOBdefaults*)*, 19
default.componentCompTable, *7, 8*, *10*
default.componentCompTable *(*LOBdefaults*)*, 19
default.LOBdbase, *3, 4*, *6*, *9, 10*, *14*
default.LOBdbase *(*LOBdefaults*)*, 19
default.oxyRanges, *9, 10*
default.oxyRanges *(*LOBdefaults*)*, 19
default.rt.windows, *3, 4*
default.rt.windows *(*LOBdefaults*)*, 19
degree_oxidation *(*LOBdbase-class*)*, 17
degree_oxidation,LOBdbase-method *(*LOBdbase-class*)*, 17
degree_oxidation<- *(*LOBdbase-class*)*, 17
degree_oxidation<-,LOBdbase-method *(*LOBdbase-class*)*, 17
doLOBscreen, 2, *7*, *10, 11*, *13–15*, *17*, *19–24*

exact_parent_neutral_mass *(*LOBdbase-class*)*, 17
exact_parent_neutral_mass,LOBdbase-method *(*LOBdbase-class*)*, 17
exact_parent_neutral_mass<- *(*LOBdbase-class*)*, 17
exact_parent_neutral_mass<-,LOBdbase-method *(*LOBdbase-class*)*, 17
exportDBtoCSV, 5
extractLOBdbasedata, 6

FA_total_no_C *(*LOBdbase-class*)*, 17
FA_total_no_C,LOBdbase-method *(*LOBdbase-class*)*, 17
FA_total_no_C<- *(*LOBdbase-class*)*, 17
FA_total_no_C<-,LOBdbase-method *(*LOBdbase-class*)*, 17
FA_total_no_DB *(*LOBdbase-class*)*, 17
FA_total_no_DB,LOBdbase-method *(*LOBdbase-class*)*, 17
FA_total_no_DB<- *(*LOBdbase-class*)*, 17
FA_total_no_DB<-,LOBdbase-method *(*LOBdbase-class*)*, 17
findIsotopes, *3, 4*
frag_ID *(*LOBdbase-class*)*, 17
frag_ID,LOBdbase-method *(*LOBdbase-class*)*, 17
frag_ID<- *(*LOBdbase-class*)*, 17
frag_ID<-,LOBdbase-method *(*LOBdbase-class*)*, 17

generateLOBdbase, *3–6*, 7, *14*, *16, 17*, *19–21*
getLOBpeaklist, *4*, 11, *24*

lipid_class *(*LOBdbase-class*)*, 17
lipid_class,LOBdbase-method *(*LOBdbase-class*)*, 17
lipid_class<- *(*LOBdbase-class*)*, 17
lipid_class<-,LOBdbase-method *(*LOBdbase-class*)*, 17
loadLOBdbase, *3*, *4*, *6*, *10*, 13, *16, 17*, *19*
LOBdbase, *4–6*, *10*, *13, 14*, 15, *17*, *19*, *21*
LOBdbase-class, 17
LOBdefaults, 19
LOBisoID_diagnostics *(*LOBSet-class*)*, 23
LOBisoID_diagnostics,LOBSet-method *(*LOBSet-class*)*, 23
LOBisoID_diagnostics<- *(*LOBSet-class*)*, 23
LOBisoID_diagnostics<-,LOBSet-method *(*LOBSet-class*)*, 23
LOBscreen_diagnostics *(*LOBSet-class*)*, 23
LOBscreen_diagnostics,LOBSet-method *(*LOBSet-class*)*, 23
LOBscreen_diagnostics<- *(*LOBSet-class*)*, 23
LOBscreen_diagnostics<-,LOBSet-method *(*LOBSet-class*)*, 23
LOBscreen_settings *(*LOBSet-class*)*, 23
LOBscreen_settings,LOBSet-method *(*LOBSet-class*)*, 23
LOBscreen_settings<- *(*LOBSet-class*)*, 23
LOBscreen_settings<-,LOBSet-method *(*LOBSet-class*)*, 23
LOBSet, *4*, *13*, 21, *21–24*
LOBSet-class, 23

mz *(*LOBdbase-class*)*, 17
mz,LOBdbase-method *(*LOBdbase-class*)*, 17
mz<- *(*LOBdbase-class*)*, 17
mz<-,LOBdbase-method *(*LOBdbase-class*)*, 17

num_compounds *(*LOBdbase-class*)*, 17
num_compounds,LOBdbase-method *(*LOBdbase-class*)*, 17
num_compounds<- *(*LOBdbase-class*)*, 17
num_compounds<-,LOBdbase-method *(*LOBdbase-class*)*, 17
num_entries *(*LOBdbase-class*)*, 17
num_entries,LOBdbase-method *(*LOBdbase-class*)*, 17
num_entries<- *(*LOBdbase-class*)*, 17
num_entries<-,LOBdbase-method *(*LOBdbase-class*)*, 17

parent_compound_name *(*LOBdbase-class*)*, 17
parent_compound_name,LOBdbase-method *(*LOBdbase-class*)*, 17
parent_compound_name<- *(*LOBdbase-class*)*, 17
parent_compound_name<-,LOBdbase-method *(*LOBdbase-class*)*, 17
parent_elem_formula *(*LOBdbase-class*)*, 17
parent_elem_formula,LOBdbase-method *(*LOBdbase-class*)*, 17
parent_elem_formula<- *(*LOBdbase-class*)*, 17
parent_elem_formula<-,LOBdbase-method *(*LOBdbase-class*)*, 17
peakdata, *21*
peakdata *(*LOBSet-class*)*, 23
peakdata,LOBSet-method *(*LOBSet-class*)*, 23
peakdata<- *(*LOBSet-class*)*, 23
peakdata<-,LOBSet-method *(*LOBSet-class*)*, 23
polarity *(*LOBSet-class*)*, 23
polarity,LOBdbase-method *(*LOBdbase-class*)*, 17
polarity,LOBSet-method *(*LOBSet-class*)*, 23
polarity<- *(*LOBSet-class*)*, 23
polarity<-,LOBdbase-method *(*LOBdbase-class*)*, 17
polarity<-,LOBSet-method *(*LOBSet-class*)*, 23

read.table, *14*

sampnames *(*LOBSet-class*)*, 23
sampnames,LOBSet-method *(*LOBSet-class*)*, 23
sampnames<- *(*LOBSet-class*)*, 23
sampnames<-,LOBSet-method *(*LOBSet-class*)*, 23
show,LOBdbase-method *(*LOBdbase-class*)*, 17

show,LOBSet-method *(LOBSet-class)*, 23
species *(LOBdbase-class)*, 17
species,LOBdbase-method
(LOBdbase-class), 17
species<- *(LOBdbase-class)*, 17
species<-,LOBdbase-method
(LOBdbase-class), 17

xcmsSet, *4*
xsAnnotate, *2*, *4*, *22*, *24*