Package 'mssearchr'

December 9, 2024

Type Package

Title Library Search Against Electron Ionization Mass Spectral Databases

Version 0.2.0

Description Perform library searches against electron ionization mass spectral databases using either the API provided by 'MS Search' software (<https://chemdata.nist.gov/dokuwiki/doku.php?id=chemdata:nistlibs>) or custom implementations of the Identity and Similarity algorithms.

License MIT + file LICENSE

URL https://mass-spec.ru/projects/gcmsdata/mssearchr/eng/

BugReports https://github.com/AndreySamokhin/mssearchr/issues

Depends R (>= 3.5.0)

Imports Rcpp

Suggests testthat (>= 3.0.0)

LinkingTo Rcpp

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

LazyData true

Author Andrey Samokhin [aut, cre, cph] (<https://orcid.org/0000-0003-0223-6087>)

Maintainer Andrey Samokhin <andrey.s.samokhin@gmail.com>

Repository CRAN

Date/Publication 2024-12-09 12:30:06 UTC

Contents

`LibrarySearch` *Perform the library search within R*

Description

Perform library search using a custom implementation of the Identity (EI Normal) or Similarity (EI Simple) algorithm. Pairwise comparison of two mass spectra is implemented in C.

Usage

```
LibrarySearch(
  msp_objs_u,
  msp_objs_l,
  algorithm = c("identity_normal", "similarity_simple"),
  search_type = c("standard", "reverse"),
  n_hits = 100L,
  hitlist_columns = c("formula", "mw", "smiles"),
  mz_min = NULL,
  mz_max = NULL,
  comments = NULL
)
```

Arguments

`msp_objs_u, msp_objs_l`
: A list of nested lists. Each nested list is a mass spectrum. Each nested list must contain at least three elements: (1) `name` (a string) - compound name (or short description); (2) `mz` (a numeric/integer vector) - m/z values of mass spectral peaks; (3) `intst` (a numeric/integer vector) - intensities of mass spectral peaks. Letters 'u' and 'l' stand for unknown and library respectively). Mass spectra should be pre-processed using the `PreprocessMassSpectra` function.

`algorithm`
: A string. Library search algorithm. Either the Identity EI Normal (`identity_normal`) or Similarity EI Simple (`similarity_simple`) algorithm.

`search_type`
: A string. Library search type: standard search (`standard`) or reverse search (`reverse`). During the standard search all peaks presented in either library or unknown spectrum are taken into account. During the reverse search all peaks that are absent in the library spectrum are ignored.

`n_hits` An integer value. The maximum number of hits (i.e., candidates) to display.

`hitlist_columns`

A character vector. Three columns are always present in the returned hitlist: `name`, `mf` or `rmf` (i.e., the match factor or the reverse match factor), and `idx` (i.e., the index of the respective library mass spectrum in the `msp_objs_l` list). Some additional columns can be added using the `hitlist_columns` argument (e.g., `cas_no`, `formula`, `inchikey`, etc.). Only scalar values (i.e., an atomic vector of unit length) are allowed.

`mz_min`, `mz_max` An integer value. Boundaries of the m/z range (all m/z values out of this range are not taken into account when the match factor is calculated).

`comments` Any R object. Some additional information. It is saved as the ’comments’ attribute of the returned list.

Value

Return a list of data frames. Each data frame is a hitlist (i.e., list of possible candidates). Each hitlist always contains three columns: `name`, `mf` or `rmf` (i.e., the match factor or the reverse match factor), and `idx` (i.e., the index of the respective library mass spectrum in the `msp_objs_l` list). Additional columns can be extracted using the `hitlist_columns` argument. Library search options are saved as the `library_search_options` attribute.

Examples

```
# Reading the 'alkanes.msp' file
msp_file <- system.file("extdata", "alkanes.msp", package = "mssearchr")

# Pre-processing
msp_objs_u <- PreprocessMassSpectra(ReadMsp(msp_file)) # unknown mass spectra
msp_objs_l <- PreprocessMassSpectra(massbank_alkanes)  # library mass spectra

# Searching using the Identity algorithm
hitlists <- LibrarySearch(msp_objs_u, msp_objs_l,
                          algorithm = "identity_normal", n_hits = 10L,
                          hitlist_columns = c("formula", "smiles", "db_no"))

# Printing a hitlist for the first compound from the 'alkanes.msp' file
print(hitlists[[1]][1:5, ])

#>        name       mf idx formula        smiles                db_no
#> 1  UNDECANE 950.5551  11  C11H24   CCCCCCCCCCC MSBNK-{...}-JP006877
#> 2  UNDECANE 928.4884  72  C11H24   CCCCCCCCCCC MSBNK-{...}-JP005760
#> 3  DODECANE 905.7546  74  C12H26  CCCCCCCCCCCC MSBNK-{...}-JP006878
#> 4 TRIDECANE 891.7862  41  C13H28 CCCCCCCCCCCCC MSBNK-{...}-JP006879
#> 5  DODECANE 885.6247  42  C12H26  CCCCCCCCCCCC MSBNK-{...}-JP005756
```

LibrarySearchUsingNistApi

Perform the library search using an API from NIST

Description

Perform the library search using an API for the MS Search software (NIST). The search is performed by calling the *nistms$.exe* file. The API is described in the NIST Mass Spectral Search Program manual. Library search options are set within the MS Search (NIST) software. To perform automatic library search the following settings should be set: (1) the 'Automatic Search On' box should be checked; (2) the 'Number of Hits to Print' field should contain reasonable value of candidates (e.g., 100).

Usage

```
LibrarySearchUsingNistApi(
  msp_objs,
  mssearch_dir = NULL,
  temp_msp_file_dir = NULL,
  overwrite_spec_list = FALSE,
  comments = NULL
)
```

Arguments

`msp_objs` A list of nested lists. Each nested list is a mass spectrum. Each nested list must contain at least three elements: (1) `name` (a string) - compound name (or short description); (2) `mz` (a numeric/integer vector) - m/z values of mass spectral peaks; (3) `intst` (a numeric/integer vector) - intensities of mass spectral peaks. The number of mass spectra should be 100 or less.

`mssearch_dir` A string. Full path to the *MSSEARCH/* directory (e.g. *C:/NIST20/MSSEARCH/*). This directory must contain the *nistms$.exe* file. If `NULL`, an attempt is made to find the path automatically using the *win.ini* file.

`temp_msp_file_dir` A string. Path to a directory where a temporary msp-file is created. If `NULL`, the temporary file is created in the *MSSEARCH/* directory

`overwrite_spec_list` A logical value. If `TRUE`, all spectra in the 'Spec List' of the MS Search (NIST) software are overwritten.

`comments` Any R object. Some additional information (e.g., library search options, the list of used libraries, etc.). It is saved as the 'comments' attribute of the returned list.

Details

The function was tested using the MS Search (NIST) software (version 2.4) and the NIST20 mass spectral database. Only two algorithms have been tested yet: 'Identity EI Normal' and 'Similarity EI Simple'.

A few temporary files are created in the *MSSEARCH/* directory according to the description provided in the NIST Mass Spectral Search Program manual.

Library search options are set within the MS Search (NIST) software. To do it, perform the following steps.

- Open the MS Search (NIST) software.
- Press the 'Library Search Options' button.
- Select the required algorithm on the 'Search' tab (e.g., 'Identity, EI Normal').
- Select the required set of libraries on the 'Libraries' tab.
- Ensure that the 'Automatic Search On' box is checked ('Automation' tab).
- Set the 'Number of Hits to Print' to reasonable value (e.g., 100) on the 'Automation' tab.
- Change other settings according to the goal (e.g., 'Presearch', 'Limits', 'Constraints', etc.).

Value

Return a list of data frames. Each data frame is a hitlist. The name of unknown compound and compound in Library Factor (InLib) are saved as the `unknown_name` and `inlib` attributes of the respective data frame. Data frames contain the following elements:

`name` A character vector. Compound name.

`mf` An integer vector. Match factor.

`rmf` An integer vector. Reverse match factor.

`prob` A numeric vector. Probability.

`lib` A character vector. Library.

`cas` A character vector. CAS number.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`id` An integer vector. ID in the database.

`ri` A numeric vector. Retention index.

Examples

```
## Not run:

# To run this example, ensure that MS Search (NIST) software is installed.

# Reading the 'alkanes.msp' file
msp_file <- system.file("extdata", "alkanes.msp", package = "mssearchr")
msp_objs <- ReadMsp(msp_file)

# Searching using the MS Search (NIST) API
```

```
hitlists <- LibrarySearchUsingNistApi(msp_objs)
print(hitlists[[1]][1:5, ])

#>           name  mf rmf  prob              lib cas formula  mw id ri
#> 1     UNDECANE 951 960 55.70 massbank_alkanes   0  C11H24 156 11  0
#> 2     UNDECANE 928 928 20.34 massbank_alkanes   0  C11H24 156 72  0
#> 3     DODECANE 906 929  8.04 massbank_alkanes   0  C12H26 170 74  0
#> 4    TRIDECANE 892 907  5.03 massbank_alkanes   0  C13H28 184 41  0
#> 5     DODECANE 886 900  3.95 massbank_alkanes   0  C12H26 170 42  0

## End(Not run)
```

massbank_alkanes *Mass spectra of alkanes*

Description

Electron ionization mass spectra of alkanes from the MassBank database (version 2023.11).

Usage

```
massbank_alkanes
```

Format

A list of nested lists. Each nested list is a mass spectrum. Each nested list contains the following elements (a more detailed description can be found in the official documentation of MassBank):

`name` A string. Name of the chemical compound analyzed.

`synon` A character vector. Alternative chemical names. The element may be absent for certain mass spectra.

`db_no` A string. Identifier of the MassBank record.

`inchikey` A string. InChIKey.

`inchi` A string. IUPAC International Chemical Identifier (InChI Code).

`smiles` A string. SMILES string

`spectrum_type` A string. MSn type of data.

`instrument_type` A string. Type of instrument.

`instrument` A string. Commercial name and manufacturer of instrument.

`ion_mode` A string. Polarity of ion detection.

`formula` A string. Chemical formula.

`mw` A string. Nominal mass.

`exactmass` A string. Exact mass.

`comments` A string. Comments.

`splash` A string. Hashed identifier of mass spectra.

`library` A string. The name and version of the database.

`mz` A numeric vector. Mass values of mass spectral peaks.

`intst` A numeric vector. Intensities of mass spectral peaks.

Source

MassBank (version 2023.11).

`PreprocessMassSpectra` *Pre-process mass spectra*

Description

Pre-process mass spectra. Pre-processing includes rounding/binning, sorting, and normalization.

Usage

```
PreprocessMassSpectra(
  msp_objs,
  bin_boundary = 0.649,
  remove_zeros = TRUE,
  max_intst = 999
)
```

Arguments

`msp_objs` A list of nested lists. Each nested list is a mass spectrum. Each nested list must contain at least three elements: (1) the `name` element (a string) - compound name (or short description); (2) the `mz` element (a numeric/integer vector) - m/z values of mass spectral peaks; (3) the `intst` (a numeric/integer vector) - intensities of mass spectral peaks.

`bin_boundary` A numeric value. The position of a bin boundary (it can be considered as a 'rounding point'). The `bin_boundary` argument must be in the following range: `0.01` <= `bin_boundary` <= `0.99`. The default value is `0.649`. This value is used in the AMDIS software and it is close to the optimal rounding rule proposed in our research (Khrisanfov, M.; Samokhin, A. A General Procedure for Rounding m/z Values in Low-resolution Mass Spectra. Rapid Comm Mass Spectrometry 2022, 36 (11), e9294. https://doi.org/10.1002/rcm.9294).

`remove_zeros` An integer value. If `TRUE`, all m/z values with zero intensity are excluded from mass spectra. It should be taken into account that all zero-intensity peaks presented in a mass spectrum are considered as 'trace peaks' in the case of MS Search software. As a result, the presence/absence of such peaks can influence the value of the match factor.

`max_intst` A numeric value. The maximum intensity (i.e., intensity of the base peak) after normalization. The default value is 999 because it is used in some electron ionization mass spectral databases including NIST.

Details

Pre-processing includes the following steps:

- Calculating a nominal mass spectrum. All floating point m/z values are rounded to the nearest integer using the value of the `bin_boundary` argument. Intensities of peaks with identical m/z values are summed.
- Intensities of mass spectral peaks are normalized to `max_intst`.
- Intensities of mass spectral peaks are rounded to the nearest integer.
- If the `remove_zeros` argument is `TRUE`, all zero-intensity peaks are removed from the mass spectrum.
- The `preprocessed` attribute is added and set to `TRUE` for the respective mass spectrum.

Value

A list of nested lists. Each nested list is a mass spectrum. Only the `mz` and `intst` elements of each nested list are modified during the pre-processing step.

Examples

```
# Original mass spectra of chlorine and methane
msp_objs <- list(
  list(name = "Chlorine",
       mz = c(34.96885, 36.96590, 69.93771, 71.93476, 73.93181),
       intst = c(0.83 * c(100, 32), c(100, 63.99, 10.24))),
  list(name = "Methane",
       mz = c(10, 11, 12, 13, 14, 15, 16, 17, 18, 19),
       intst = c(0, 0, 25, 75, 155, 830, 999, 10, 0, 0))
)
matrix(c(msp_objs[[1]]$mz, msp_objs[[1]]$intst), ncol = 2) # Chlorine
matrix(c(msp_objs[[2]]$mz, msp_objs[[2]]$intst), ncol = 2) # Methane

# Pre-processed mass spectra of chlorine and methane
pp_msp_objs <- PreprocessMassSpectra(msp_objs, remove_zeros = TRUE)
matrix(c(pp_msp_objs[[1]]$mz, pp_msp_objs[[1]]$intst), ncol = 2) # Chlorine
matrix(c(pp_msp_objs[[2]]$mz, pp_msp_objs[[2]]$intst), ncol = 2) # Methane
```

ReadMsp *Read mass spectra from an msp-file (NIST format)*

Description

Read an msp-file containing mass spectra in the NIST format. The complete description of the format can be found in the NIST Mass Spectral Search Program manual. A summary is presented below in the "Description of the NIST format" section.

Usage

```
ReadMsp(input_file)
```

Arguments

`input_file`	A string. The name of a file.

Details

Data from an msp-file are read without any modification (e.g., the order of mass values is not changed, zero-intensity peaks are preserved, etc.).

Value

Return a list of nested lists. Each nested list is a mass spectrum. Almost all metadata fields (e.g., "Name", "CAS#", "Formula", "MW", etc.) are represented as strings. All "Synon" fields are merged into a single character vector. Mass values and intensities are represented as numeric vectors (`mz` and `intst`). Names of fields are slightly modified:

- names are converted to lowercase;
- hash symbols are replaced with `_no`;
- any other special character is replaced with an underscore character.

Description of the NIST format

The summary was prepared using the NIST Mass Spectral Search Program manual v.2.4 (2020).

- An msp-file can contain as many spectra as wanted.
- Each spectrum must start with the "Name" field. There must be something in this field.
- The "Num Peaks" field is also required. It must contain the number of mass/intensity pairs.
- Some optional fields (e.g. "Comments", "Formula", "MW") can be between the "Name" and "Num Peaks" fields.
- When a spectrum is exported from the NIST library it also contains the "NIST#" and "DB#" fields. The "NIST#" field is on the same line as the "CAS#" field and separated by a semicolon.
- Each field should be on a separate line (the "NIST#" field is an exception from this rule)
- The mass/intensity list begins on the line following the "Num Peaks" field. The peaks need not be normalized, and the masses need not be ordered. The exact spacing and delimiters used for the mass/intensity pairs are unimportant. The following characters are accepted as delimiters: `'space'`, `'tab'`, `','`, `';'`, `':'`. Parentheses, square brackets and curly braces (`'('`, `'('`, `'['`, `']'`, `'{'`, and `'}'`) are also allowed.
- The "Name" field can be up to 511 characters.
- The "Comments" field can be up to 1023 characters.
- The "Formula" field can be up to 23 characters.
- The "Synon" field may be repeated.

Examples

```
# Reading the 'alkanes.msp' file
msp_file <- system.file("extdata", "alkanes.msp", package = "mssearchr")
msp_objs <- ReadMsp(msp_file)

# Plotting the first mass spectrum from the 'msp_objs' list
par_old <- par(yaxs = "i")
plot(msp_objs[[1]]$mz, msp_objs[[1]]$intst,
     ylim = c(0, 1000), main = msp_objs[[1]]$name,
     type = "h", xlab = "m/z", ylab = "Intensity", bty = "l")
par(par_old)
```

WriteMsp *Write mass spectra in an msp-file (NIST format)*

Description

Write mass spectra in an msp-file (NIST format).

Usage

```
WriteMsp(msp_objs, output_file, fields = NULL)
```

Arguments

msp_objs	A list of nested lists. Each nested list is a mass spectrum. Each nested list must contain at least three elements: (1) name (a string) - compound name (or short description); (2) mz (a numeric/integer vector) - m/z values of mass spectral peaks; (3) intst (a numeric/integer vector) - intensities of mass spectral peaks.
output_file	A string. The name of a file.
fields	A character vector. Names of elements in an R list (not the original field names from an msp-file) to be exported. For example, if only CAS number is needed to be exported, the 'cas_no' (not 'cas#') should be passed. If NULL, all fields are exported. The output file always contains the 'Name' field, the 'Num Peaks' field, and the mass/intensity list.

Details

Names of all fields are exported in lower case. It does not cause any problem in the case of the MS Search (NIST) software (however correct operation with other software products has not been tested). Only in a few cases hash symbols and spaces are restored:

- the cas_no element is exported as the 'cas#' field;
- the nist_no element is exported as the 'nist#' field;
- the num_peaks element is exported as the 'num peaks' field.

Value

NULL is returned.

Examples

```
# Exporting mass spectra
# Only 'Name', 'SMILES', 'Formula', and 'Num Peaks' fields are exported.
WriteMsp(massbank_alkanes[1:3], ”test.msp”, fields = c(”smiles”, ”formula”))
```

Index