Package 'mspepsearchr'

November 9, 2025

Type Package

Title Batch Mass Spectral Library Searches Using the MSPepSearch (NIST) Tool

Version 0.2.0

Description A convenient interface to perform batch searches of mass spectral data against NIST-format databases using the MSPepSearch tool developed by NIST. An R function automatically constructs search commands, calls the MSPepSearch executable, parses the output, and represents the library search results as an R object.

License MIT + file LICENSE

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

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

Encoding UTF-8

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

Imports mssearchr,
parallel,
utils

VignetteBuilder knitr

Contents

`IdentitySearchEiNormal` *Library search using the 'Identity EI Normal' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Identity EI Normal' algorithm.

Uses an extensive set of prescreens - a total of four different prescreen criteria. Note that if your spectrum is not passed through prescreen it cannot be found in the hit list. For spectra that have no closely matching spectra in the library, it may be possible to improve results by searching without a prescreen.

Usage

```
IdentitySearchEiNormal(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  ri_column_type = "stdnp",
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.

`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries:
	`c("C:/NIST23/MSSEARCH/mainlib/",`
	`  "C:/NIST23/MSSEARCH/replib/")`

`n_threads`	An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

presearch A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

n_hits An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

search_method A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`ri_column_type` A string or `NULL`. The chromatographic column type. If `NULL`, retention indices are not added to the hitlist. This argument is ignored, if the `/RI` flag is set manually via `addl_cli_args`.

StdNP (`"stdnp"` **or** `"n"`) Standard non-polar (e.g., DB-1).

SemiStdNP(`"semistdnp"` **or** `"s"`) Semi-standard non-polar (e.g., DB-5).

StdPolar(`"stdpolar"` **or** `"p"`) Standard polar (e.g., DB-WAX).

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound, its InChIKey, and compound in Library Factor (InLib) are stored as the attributes `unknown_name`, `unknown_inchikey`, and `inlib` respectively. 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.

`pss_mf` An integer vector. PSS match factor.

`prob` A numeric vector. Probability.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`ri` A numeric vector. Retention index.

`ri_type` A character vector. RI type.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra", "alkanes_ei_lr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_ei_lr",
                        package = "mspepsearchr")

# Library searching
hitlists <- IdentitySearchEiNormal(msp_path, lib_path,
                                   best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>        name  mf rmf pss_mf prob formula  mw exact_mass ...
#> 1  UNDECANE 897 931    897 43.5  C11H24 156    156.188 ...
#> 2  DODECANE 883 917    898 28.4  C12H26 170    170.203 ...
#> 3 TRIDECANE 875 897    896 22.6  C13H28 184    184.219 ...
```

`IdentitySearchHighRes` *Library search using the 'Identity HighRes NoPrecursor' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Identity HiRes NoPrecursor' (also known as 'Identity In-source HiRes') algorithm.

Identity HiRes No Precursor (Formerly named In-source/EI with accurate ion m/z) - Searches for an HiRes No Precursor or MS/MS spectrum in a library containing HiRes No Precursor or MS/MS spectra. The library must be built with Lib2NIST ver. 1.0.4.28 (07/05/2013) or later and have files peak_em0.inu and peak_em0.dbu. Unlike MS/MS search, this search does not compare precursor m/z values. Currently, HiRes No Precursor spectra added with NIST MS Search to a user library may be searched with HiRes No Precursor search only with Presearch OFF option.

Usage

```
IdentitySearchHighRes(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  mz_tolerance = list(value = 0.01, units = "mz"),
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  ri_column_type = "stdnp",
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.
`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries: `c("C:/NIST23/MSSEARCH/mainlib/", "C:/NIST23/MSSEARCH/replib/")`
`n_threads`	An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.
`presearch`	A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`mz_tolerance` A list with two elements: a numeric tolerance value and a character string specifying the units. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`.

`n_hits` An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method` A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`ri_column_type` A string or `NULL`. The chromatographic column type. If `NULL`, retention indices are not added to the hitlist. This argument is ignored, if the `/RI` flag is set manually via `addl_cli_args`.

StdNP (`"stdnp"` **or** `"n"`) Standard non-polar (e.g., DB-1).

SemiStdNP(`"semistdnp"` **or** `"s"`) Semi-standard non-polar (e.g., DB-5).

StdPolar(`"stdpolar"` **or** `"p"`) Standard polar (e.g., DB-WAX).

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound and its InChIKey are stored as the attributes `unknown_name` and `unknown_inchikey` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`score` An integer vector. Match factor.

`dot` An integer vector. Dot product.

`rdot` An integer vector. Reverse match factor.

`pss_dot` An integer vector. PSS match factor.

`prob` A numeric vector. Probability.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`ri` A numeric vector. Retention index.

`ri_type` A character vector. RI type.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra",
                        "chlorine_compounds_ei_hr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "hrei_msdb_subset_ei_hr",
                        package = "mspepsearchr")

# Library searching
hitlists <- IdentitySearchHighRes(msp_path, lib_path,
                                  best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>                        name score dot rdot pss_dot prob  formula  mw ...
#> 1 Hexachlorocyclopentadiene   830 862  879     876 98.6    C5Cl6 270 ...
#> 2         Pentachlorophenol   166 448  473     755  1.0  C6HCl5O 264 ...
#> 3     2,4,6-Trichlorophenol    57 217  309     372  0.1 C6H3Cl3O 196 ...
```

`IdentitySearchMsMs` *Library search using the 'Identity MS/MS' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Identity MS/MS' algorithm.

Searches for a MS/MS spectrum in a library of MS/MS spectra. This may be used for high resolution spectrum searching.

Usage

```
IdentitySearchMsMs(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  precursor_ion_tol = list(value = 20, utits = "ppm"),
  product_ions_tol = list(value = 0.01, units = "mz"),
  ignore_precursor_ion_tol = list(value = 1.6, units = "mz"),
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra` Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.

`libraries` A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries:

```
c("C:/NIST23/MSSEARCH/mainlib/",
  "C:/NIST23/MSSEARCH/replib/")
```

`n_threads` An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
```

```
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`precursor_ion_tol`
: A list with two elements: a numeric tolerance value and a character string specifying the units. Precursor m/z tolerance. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`.

`product_ions_tol`
: A list with two elements: a numeric tolerance value and a character string specifying the units. Product ion m/z tolerance. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`.

`ignore_precursor_ion_tol`
: A list with two elements: a numeric tolerance value and a character string specifying the units. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`. Mass spectral peaks within the specified interval around the precursor are ignored. If `NULL`, the tolerance is set as the sum of `precursor_ion_tol` and `product_ions_tol`.

`n_hits`
: An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method`
: A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound, its InChIKey, and precursor m/z are stored as the attributes `unknown_name`, `unknown_inchikey`, and `prec_mz` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`score` An integer vector. Match factor.

`dot` An integer vector. Dot product.

`rdot` An integer vector. Reverse match factor.

`pss_dot` An integer vector. PSS match factor.

`prob` A numeric vector. Probability.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`charge` An integer vector. Precursor ion charge.

`prec_type` A character vector. Precursor ion type.

`delta_mz` A numeric vector. Precursor m/z difference.

`prec_mz` A numeric vector. Precurson ion m/z.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra",
                        "mw288_compounds_msms_hr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_msms_hr",
                        package = "mspepsearchr")

# Library searching
hitlists <- IdentitySearchMsMs(
  spectra = msp_path,
  libraries = lib_path,
  best_hits_only = TRUE,
  precursor_ion_tol = list(value = 0.1, utits = "mz"),
  product_ions_tol = list(value = 0.01, units = "mz")
)

# Printing the top three rows of the first hitlist
print(head(hitlists[[2]], 3L))

#>                             name score dot rdot pss_dot prob ...
#> 1                  Testosterone   920 957  972     965   99 ...
#> 2               Epitestosterone   583 762  813     794    1 ...
#> 3 Dehydroepiandrosterone (DHEA)   192 366  398     451    0 ...
```

`OpenHelpFile` *Open MSPepSearch help file*

Description

Open the *MSPepSearch64.exe.hlp.txt* file included with the MSPepSearch (NIST) tool in the default text editor.

Usage

```
OpenHelpFile()
```

Value

Return NULL.

`SimilaritySearchEiHybrid` *Library search using the 'Similarity EI Hybrid' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Similarity EI Hybrid' algorithm.

Hybrid searching uses both the logic of normal searching plus the logic of neutral loss searching. It uses the same algorithm as MS/MS Hybrid search with unit charges and Nominal MW instead of Precursor m/z. Older EI libraries may be indexed for Hybrid (EI) search using Tools / (Re) Index EI Hybrid Search menu item.

Usage

```
SimilaritySearchEiHybrid(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  nominal_mw = NULL,
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  ri_column_type = "stdnp",
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

spectra	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.

`libraries` A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries:

```
c("C:/NIST23/MSSEARCH/mainlib/",
  "C:/NIST23/MSSEARCH/replib/")
```

`n_threads` An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`nominal_mw` An integer value. The nominal molecular weight, applied to all spectra. When `NULL`, nominal molecular weight is extracted from the corresponding metadata field of in the MSP file.

`n_hits` An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method` A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`ri_column_type` A string or `NULL`. The chromatographic column type. If `NULL`, retention indices are not added to the hitlist. This argument is ignored, if the `/RI` flag is set manually via `addl_cli_args`.

StdNP (`"stdnp"` **or** `"n"`) Standard non-polar (e.g., DB-1).

SemiStdNP(`"semistdnp"` **or** `"s"`) Semi-standard non-polar (e.g., DB-5).

StdPolar(`"stdpolar"` **or** `"p"`) Standard polar (e.g., DB-WAX).

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound and its InChIKey are stored as the attributes `unknown_name` and `unknown_inchikey` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`mf` An integer vector. Match factor (EI mass spectra).

`rmf` An integer vector. Reverse match factor (EI mass spectra).

`pss_mf` An integer vector. PSS match factor (EI mass spectra).

`deltamass` A numeric vector. Delta mass.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`o_match` An integer vector.

`o_r_match` An integer vector.

`o_pss_match` An integer vector.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra", "alkanes_ei_lr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_ei_lr",
                        package = "mspepsearchr")

# Library searching
hitlists <- SimilaritySearchEiHybrid(msp_path, lib_path,
                                     best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>          name  mf rmf pss_mf deltamass formula  mw exact_mass ...
#> 1   TRIDECANE 945 975    945  -28.0313  C13H28 184    184.219 ...
#> 2 TETRADECANE 945 971    945  -42.0470  C14H30 198    198.235 ...
#> 3    UNDECANE 945 961    948    0.0000  C11H24 156    156.188 ...
```

`SimilaritySearchEiNeutralLoss` *Library search using the 'Similarity EI Neutral Loss' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Similarity EI Neutral Loss' algorithm.

The neutral losses from the molecular ion are used to produce a neutral loss spectrum. In many cases, this type of matching will provide structurally similar matches.

Usage

```
SimilaritySearchEiNeutralLoss(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  nominal_mw = NULL,
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  ri_column_type = "stdnp",
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.
`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries: `c("C:/NIST23/MSSEARCH/mainlib/", "C:/NIST23/MSSEARCH/replib/")`
`n_threads`	An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`nominal_mw` An integer value. The nominal molecular weight, applied to all spectra. When `NULL`, nominal molecular weight is extracted from the corresponding metadata field of in the MSP file.

`n_hits` An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method` A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`ri_column_type` A string or `NULL`. The chromatographic column type. If `NULL`, retention indices are not added to the hitlist. This argument is ignored, if the `/RI` flag is set manually via `addl_cli_args`.

StdNP (`"stdnp"` **or** `"n"`) Standard non-polar (e.g., DB-1).

SemiStdNP(`"semistdnp"` **or** `"s"`) Semi-standard non-polar (e.g., DB-5).

StdPolar(`"stdpolar"` **or** `"p"`) Standard polar (e.g., DB-WAX).

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown

compound and its InChIKey are stored as the attributes `unknown_name` and `unknown_inchikey` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`mf` An integer vector. Match factor (EI mass spectra).

`rmf` An integer vector. Reverse match factor (EI mass spectra).

`pss_mf` An integer vector. PSS match factor (EI mass spectra).

`deltamass` A numeric vector. Delta mass.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`ri` A numeric vector. Retention index.

`ri_type` A character vector. RI type.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra", "alkanes_ei_lr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_ei_lr",
                        package = "mspepsearchr")

# Library searching
hitlists <- SimilaritySearchEiNeutralLoss(msp_path, lib_path,
                                          best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>        name  mf rmf pss_mf deltamass formula  mw exact_mass ...
#> 1  UNDECANE 945 961    948         0  C11H24 156    156.188 ...
#> 2  DODECANE 895 933    899       -14  C12H26 170    170.203 ...
#> 3 TRIDECANE 815 912    815       -28  C13H28 184    184.219 ...
```

`SimilaritySearchEiSimple` *Library search using the 'Similarity EI Simple' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Similarity EI Simple' algorithm.

In the Similarity search, a similar set of four prescreen techniques are used but the final match factor is calculated without using m/z weighting. In general this is more likely to produce spectra that are from molecules structurally similar to the compound that produced the submitted spectrum.

Usage

```
SimilaritySearchEiSimple(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  ri_column_type = "stdnp",
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.
`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries: `c("C:/NIST23/MSSEARCH/mainlib/", "C:/NIST23/MSSEARCH/replib/")`
`n_threads`	An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`n_hits` An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method` A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`ri_column_type` A string or `NULL`. The chromatographic column type. If `NULL`, retention indices are not added to the hitlist. This argument is ignored, if the `/RI` flag is set manually via `addl_cli_args`.

StdNP (`"stdnp"` **or** `"n"`) Standard non-polar (e.g., DB-1).

SemiStdNP(`"semistdnp"` **or** `"s"`) Semi-standard non-polar (e.g., DB-5).

StdPolar(`"stdpolar"` **or** `"p"`) Standard polar (e.g., DB-WAX).

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound and its InChIKey are stored as the attributes `unknown_name` and `unknown_inchikey` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`mf` An integer vector. Match factor (EI mass spectra).

`rmf` An integer vector. Reverse match factor (EI mass spectra).

`pss_mf` An integer vector. PSS match factor (EI mass spectra).

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`ri` A numeric vector. Retention index.

`ri_type` A character vector. RI type.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra", "alkanes_ei_lr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_ei_lr",
                        package = "mspepsearchr")

# Library searching
hitlists <- SimilaritySearchEiSimple(msp_path, lib_path,
                                     best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>        name  mf rmf pss_mf formula  mw exact_mass ...
#>    UNDECANE 945 961    948  C11H24 156    156.188 ...
#>   TRIDECANE 923 956    938  C13H28 184    184.219 ...
#> TETRADECANE 917 945    939  C14H30 198    198.235 ...
```

`SimilaritySearchMsmsHybrid` *Library search using the 'Similarity MS/MS Hybrid' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Similarity MS/MS Hybrid' algorithm.

Search for an MS/MS spectrum in a library of MS/MS spectra. Uses the logic of normal searching and the logic of neutral loss searching. Only library spectra with the same precursor charge as that of the search spectrum are considered. If a search spectrum precursor charge is missing, it is set to +1. All product ion charges are set to 1 or 2 in case precursor charge is 2 or greater. Neutral loss searching is done after changing library spectrum peaks' m/z values so that they have the same neutral losses with respect to the search spectrum precursor as they had with respect to the library spectrum precursor in the original library spectrum. This changed spectrum is compared to the search spectrum; found peak matches are included in the dot product. If a library spectrum peak matches different search spectrum peaks in both direct and neutral loss searches, its intensity is split between these two matches to maximize the dot product and keep total library spectrum intensity unchanged.

Usage

```
SimilaritySearchMsmsHybrid(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  precursor_ion_tol = list(value = 20, utits = "ppm"),
  product_ions_tol = list(value = 0.01, units = "mz"),
  ignore_precursor_ion_tol = NULL,
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.
`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries: `c("C:/NIST23/MSSEARCH/mainlib/", "C:/NIST23/MSSEARCH/replib/")`
`n_threads`	An integer value. Number of parallel threads to use for computation. External

process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`precursor_ion_tol`

A list with two elements: a numeric tolerance value and a character string specifying the units. Precursor m/z tolerance. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`.

`product_ions_tol`
: A list with two elements: a numeric tolerance value and a character string specifying the units. Product ion m/z tolerance. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`.

`ignore_precursor_ion_tol`
: A list with two elements: a numeric tolerance value and a character string specifying the units. Valid examples include `list(0.01, "mz")` and `list(50, "ppm")`. Mass spectral peaks within the specified interval around the precursor are ignored. If `NULL`, the tolerance is set as the sum of `precursor_ion_tol` and `product_ions_tol`.

`n_hits`
: An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method`
: A string. Search method and the preferred order of spectra in the hit list.

  Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

  Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

  PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only`
: A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance`
: An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min`
: An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max`
: An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`load_in_memory`
: A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir`
: A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args`
: A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound, its InChIKey, and precursor m/z are stored as the attributes `unknown_name`, `unknown_inchikey`, and `prec_mz` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`score` An integer vector. Match factor.

`dot` An integer vector. Dot product.

`rdot` An integer vector. Reverse match factor.

`pss_dot` An integer vector. PSS match factor.

`deltamass` A numeric vector. Delta mass.

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`charge` An integer vector. Precursor ion charge.

`prec_type` A character vector. Precursor ion type.

`delta_mz` A numeric vector. Precursor m/z difference.

`prec_mz` A numeric vector. Precurson ion m/z.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

`o_score` An integer vector.

`o_dotprod` An integer vector.

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra",
                        "mw288_compounds_msms_hr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_msms_hr",
                        package = "mspepsearchr")

# Library searching
hitlists <- SimilaritySearchMsmsHybrid(msp_path, lib_path,
                                       best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[2]], 3L))

#>                     name score dot rdot pss_dot deltamass  formula ...
```

```
#> 1            Testosterone   959 965  976     970    0.0000 C19H28O2 ...
#> 2 4-Androstene-3,17-dione   951 957  970     962    2.0156 C19H26O2 ...
#> 3            Progesterone   935 955  958     980  -26.0157 C21H30O2 ...
```

`SimilaritySearchMsMsInEi` *Library search using the 'Similarity MS/MS in EI' algorithm*

Description

Perform library searches against electron ionization mass spectral databases using the 'Similarity MS/MS in EI' algorithm.

Search for a MS/MS spectrum in a library of EI spectra.

Usage

```
SimilaritySearchMsMsInEi(
  spectra,
  libraries,
  n_threads = 1L,
  presearch = "default",
  nominal_mw = NULL,
  n_hits = 100L,
  search_method = "standard",
  best_hits_only = FALSE,
  min_abundance = 1L,
  mz_min = NULL,
  mz_max = NULL,
  load_in_memory = TRUE,
  temp_dir = NULL,
  addl_cli_args = NULL
)
```

Arguments

`spectra`	Mass spectra to search. It can be either a string giving the path to an MSP file or a list of nested lists, where each nested list represents one 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. An object of this format can be created manually, or read from an MSP file with the `ReadMsp` function.
`libraries`	A character vector. Paths to mass spectral libraries stored in the NIST format. For example, to search against the `mainlib` and `replib` libraries: `c("C:/NIST23/MSSEARCH/mainlib/", "C:/NIST23/MSSEARCH/replib/")`
`n_threads`	An integer value. Number of parallel threads to use for computation. External process parallelization is employed, where each R worker launches an independent CLI call.

`presearch` A string or a list. The presearch routine applies simple algorithms to reduce the search space and significantly speed up search times. However, note that due to inherent blind spots in these algorithms, even spectra that closely match may sometimes be excluded.

Default Normal operation, use presearch screening before spectrum-by-spectrum match factor calculation. Acceptable values include:

```
"default"
list("default")
list(type = "default")
```

Fast Use almost the same presearch screening to select on average 3 times less spectra than in Default. Acceptable values include:

```
"fast"
list("fast")
list(type = "fast")
```

Off Search the entire database using only the detailed match algorithm. This option can take much longer and may be helpful when no close matches are obtained using the default search. Acceptable values include:

```
"off"
list("off")
list(type = "off")
```

MW Restrict search to only those molecules with a user supplied molecular weight. Use this when the molecular weight of a substance is known. Acceptable values include:

```
list("mw", integer(1L))
list(type = "mw", mw = integer(1L))
```

InChIKey Restrict search to only molecules with the first segment (i.e., `n_segments = 1L`) of the InChIKey being the same as that of the search spectrum. Use this presearch when the chemical structure or InChIKey of the search spectrum is known. To use this presearch, a library must be indexed by InChIKey. An older library may be indexed by selecting (Re)Index InChIKey from the Tools menu of the MS Search (NIST) software. The search and found compounds do not have to have mass spectral peaks: hits with score=0 are included in the hit list. This allows to search `nist_ri` library. To retain all hits (up to 400) use Identity Normal, Quick, or Similarity Simple search. Other searches impose certain conditions on the library spectra thus excluding hits. The number of identical segments in InChIKey strings can be set to 1, 2, or 3. Acceptable values include:

```
list("inchikey", integer(1L))
list(type = "inchikey", n_segments = integer(1L))
```

`nominal_mw` An integer value. The nominal molecular weight, applied to all spectra. When `NULL`, nominal molecular weight is extracted from the corresponding metadata field of in the MSP file.

`n_hits` An integer value. The number of hits to return (up to 100 or 400, depending on the search and presearch algorithms used).

`search_method` A string. Search method and the preferred order of spectra in the hit list.

Standard (`"standard"`) All peaks presented in either library or unknown spectrum are taken into account (full spectrum search).

Reverse (`"reverse"`) Peaks in the target spectrum that are absent in the library spectrum are ignored (impurity tolerant search

PSS (`"pss"`) Peaks in the library spectrum that are absent in the target spectrum are ignored (partial spectrum search).

`best_hits_only` A logical value. If `TRUE`, only the best matching spectrum for each compound is displayed. For the MSPepSearch tool, CAS numbers are required to ensure a single hit per compound. In contrast, MS Search can use not only CAS number but also chemical names to remove duplicates from the hitlist.

`min_abundance` An integer value. The smallest peak that will be used in comparison. Used to force small peaks in the target spectrum to be ignored. Specify minimum abundance on the basis of 999; accepted values are from 1 to 999, where 2 means 0.2% of base peak intensity, 50 means 5%, etc. (Base peak = 999).

`mz_min` An integer value or `NULL`. Used to determine the lowest m/z used for match factor calculation. Without this specification, the matching algorithm starts comparison at the higher of the minimum m/z values in the target (search) or the library spectra. This is done in order to attempt to ignore peaks in one spectrum that do not match peaks in the other simply because the starting m/z was higher. For instance, if the library spectrum began at m/z 20 and the target began at m/z 40, library peaks between 20-40 m/z could cause the match factor to be seriously penalized. This approach, however, can produce artificially high match factors when either the library or search spectrum are missing low m/z peaks (partial spectra). To avoid this problem a minimum mass may be specified - this should generally be the minimum m/z from the instrument that produced the spectrum for library searching.

`mz_max` An integer value or `NULL`. Peaks above the specified mass are ignored. Use this to exclude spurious high-mass peaks in the search spectrum.

`load_in_memory` A logical value. When `TRUE`, up to 16 libraries are loaded in memory (a 2 GB limit applies to the 32-bit version).

`temp_dir` A string or `NULL`. Path to a directory where temporary files are created. If `NULL`, the `tempdir` function is used to determine the temp directory.

`addl_cli_args` A character vector or `NULL`. Additional arguments passed directly to the MSPepSearch tool via the command-line interface. Use with caution, as only a basic check is performed: an error is returned if a specific argument is duplicated. However, this check is not comprehensive, it does not treat aliases as equivalent arguments, nor does it verify that the provided arguments are compatible with each other.

Value

Return a list of data frames. Attributes of the list contain supplementary information, including the executed command, time of the call, and performance-related metadata extracted from the output file. Each data frame represents a hit list. Within each data frame, the name of the unknown compound and its InChIKey are stored as the attributes `unknown_name` and `unknown_inchikey` respectively. Data frames contain the following elements:

`name` A character vector. Compound name.

`mf` An integer vector. Match factor (EI mass spectra).

`rmf` An integer vector. Reverse match factor (EI mass spectra).

`pss_mf` An integer vector. PSS match factor (EI mass spectra).

`formula` A character vector. Chemical formula.

`mw` An integer vector. Molecular weight.

`exact_mass` A numeric vector. Exact mass.

`inchikey` A character vector. InChIKey hash.

`cas` A character vector. CAS number.

`lib` A character vector. Library.

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

Note

This documentation includes content adapted from the official MS Search (NIST) help manual.

Examples

```
# Setting paths to an MSP file and mass spectral library
msp_path <- system.file("extdata", "spectra",
                        "alkyl_fragments_msms_hr.msp",
                        package = "mspepsearchr")
lib_path <- system.file("extdata", "libraries", "massbank_subset_ei_lr",
                        package = "mspepsearchr")

# Library searching
hitlists <- SimilaritySearchMsMsInEi(msp_path, lib_path,
                                     best_hits_only = TRUE)

# Printing the top three rows of the first hitlist
print(head(hitlists[[1]], 3L))

#>        name  mf rmf pss_mf formula  mw exact_mass ...
#> 1  DODECANE 262 557    923  C12H26 170    170.203 ...
#> 2  UNDECANE 260 554    926  C11H24 156    156.188 ...
#> 3 TRIDECANE 242 515    921  C13H28 184    184.219 ...
```

Index