

Package: SpectraToQueries (via r-universe)

October 13, 2026

Title Spectra to queries

Version 0.0.0.9002

Maintainer Adriano Rutz <adafede@gmail.com>

Description SpectraToQueries provides the infrastructure to translate spectra to queries.

License AGPL (>= 3)

URL <https://github.com/spectra-to-knowledge/spectra-to-queries>,
<https://spectra-to-knowledge.github.io/SpectraToQueries>

BugReports <https://github.com/spectra-to-knowledge/spectra-to-queries/issues>

Depends R (>= 4.5.0)

Imports MsBackendMgf (>= 1.18.0), Spectra (>= 1.20.1), tidytable (>= 0.11.2)

Suggests testthat (>= 3.3.2)

ByteCompile TRUE

Collate 'SpectraToQueries-package.R' 'combine_ions_minimal.R'
'create_dir.R' 'create_matrix.R' 'data.R' 'filter_matrix.R'
'fix_binned_mzs.R' 'generate_combinations.R' 'globals.R'
'harmonize_mzs.R' 'normalize_peaks.R'
'perform_list_of_queries.R' 'progress_utils.R'
'spectra_to_queries.R' 'spectra_utils.R'

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData TRUE

LazyDataCompression bzip2

biocViews knowledgeExtraction, spectralInformation, queryingSystem

X-schema.org-keywords knowledgeExtraction, spectralInformation, queryingSystem
Config/pak/sysreqs cmake make libuv1-dev
Repository https://spectra-to-knowledge.r-universe.dev
Date/Publication 2026-10-13 18:59:41 UTC
RemoteUrl https://github.com/spectra-to-knowledge/SpectraToQueries
RemoteRef main
RemoteSha b2a2560f89ad029b7e9b3541685e9d862d2f3066

Contents

combine_ions_minimal	2
contains_all_mz	3
contains_all_mz_tol	3
create_dir	4
create_matrix	4
filter_matrix	5
fix_binned_mzs	5
generate_combinations	6
generate_combinations_progress	7
harmonize_mzs	7
mia_spectra_df	8
mia_spectra_grouped_df	9
normalize_peaks	10
perform_list_of_queries	11
perform_list_of_queries_progress	11
perform_query	12
prepare_query_data	13
spectra_to_queries	13
Index	15

combine_ions_minimal	<i>Combine ions into minimal AND/OR expression with sorting</i>
----------------------	---

Description

Combine ions into minimal AND/OR expression with sorting

Usage

combine_ions_minimal(ion_lists)

Arguments

ion_lists List of character vectors representing ion combinations

Value

Single character string representing nested AND/OR expression

contains_all_mz	<i>Check if spectrum contains all target m/z values</i>
-----------------	---

Description

Check if spectrum contains all target m/z values

Usage

```
contains_all_mz(spec_mz, target_mz, dalton, ppm)
```

Arguments

spec_mz	Numeric vector of spectrum m/z values
target_mz	Numeric vector of target m/z values to find
dalton	Tolerance in Dalton
ppm	Tolerance in parts per million

Value

Logical indicating if all targets are found

Examples

```
NULL
```

contains_all_mz_tol	<i>Check if spectrum contains all target m/z values (internal variant with pre-computed tolerances)</i>
---------------------	---

Description

Check if spectrum contains all target m/z values (internal variant with pre-computed tolerances)

Usage

```
contains_all_mz_tol(spec_mz, target_mz, tol)
```

Arguments

spec_mz	Numeric vector of spectrum m/z values
target_mz	Numeric vector of target m/z values to find
tol	Numeric vector of pre-computed tolerances (length must match target_mz)

Value

Logical indicating if all targets are found

Examples

```
NULL
```

create_dir	<i>Create directory</i>
------------	-------------------------

Description

This function creates a directory at the specified path if it does not already exist.

Usage

```
create_dir(export)
```

Arguments

export	Path to the directory to be created
--------	-------------------------------------

Value

Message indicating the status of directory creation

Source

https://github.com/taxonomicallyinformedannotation/tima/blob/main/R/create_dir.R

create_matrix	<i>Create matrix</i>
---------------	----------------------

Description

Create matrix

Usage

```
create_matrix(spectra, zero_val = 0, name)
```

Arguments

spectra	Spectra
zero_val	Zero value
name	Name

Value

A matrix with spectra intensity data

Examples

NULL

filter_matrix	<i>Filter matrix</i>
---------------	----------------------

Description

Filter matrix

Usage

filter_matrix(matrix, n)

Arguments

- | | |
|--------|--|
| matrix | Matrix to filter |
| n | Minimum number of non-zero values per column |

Value

Filtered matrix with columns having at least n non-zero values

Examples

NULL

fix_binned_mzs	<i>Fix binned mzs</i>
----------------	-----------------------

Description

Fix binned mzs

Usage

fix_binned_mzs(binned_m, original_mzs, dalton, ppm, decimals)

Arguments

<code>binned_m</code>	Binned matrix
<code>original_mzs</code>	Original m/z values from spectra
<code>dalton</code>	Tolerance in Dalton
<code>ppm</code>	Tolerance in parts per million
<code>decimals</code>	Number of decimal places for rounding

Value

Matrix with corrected m/z column names

Examples

NULL

`generate_combinations` *Generate combinations*

Description

Generate combinations

Usage

```
generate_combinations(x, max_ions)
```

Arguments

<code>x</code>	Vector of elements to combine
<code>max_ions</code>	Maximum number of ions in combinations

Value

List of all combinations

Examples

NULL

`generate_combinations_progress`*Generate combinations with progress*

Description

Generate combinations with progress

Usage

```
generate_combinations_progress(  
  indices,  
  ions_list,  
  max_ions,  
  show_progress = TRUE  
)
```

Arguments

<code>indices</code>	Vector of indices to process
<code>ions_list</code>	List of ions for each index
<code>max_ions</code>	Maximum number of ions in combinations
<code>show_progress</code>	Logical: show progress bar (default TRUE)

Value

List of all combinations for each index

Examples

```
NULL
```

`harmonize_mzs`*Harmonize mzs*

Description

Harmonize mzs

Usage

```
harmonize_mzs(spectra, dalton, ppm)
```

Arguments

spectra	Spectra object
dalton	Tolerance in Dalton
ppm	Tolerance in parts per million

Value

Spectra with harmonized m/z values

Examples

NULL

mia_spectra_df	<i>Example mass spectrometry spectra from MIADB (as portable data.frame)</i>
----------------	--

Description

A data.frame containing 321 spectra of monoterpene indole alkaloids (MIAs) from the MIADB (Monoterpene Indole Alkaloid Database). This is stored as a portable data.frame with m/z and intensity peaks as list columns, and is automatically converted to a Spectra object when used in spectra_to_queries().

Usage

mia_spectra_df

Format

A data.frame with 321 rows and 37 columns:

- **Metadata columns:** msLevel, rtime, acquisitionNum, TITLE, PRECURSOR_MZ, SKELETON, and other MS/MS spectrum attributes
- **mz (list):** Fragment m/z values for each spectrum
- **intensity (list):** Fragment intensities for each spectrum

Total of approximately 48,510 peaks across all spectra.

Details

The mia_spectra_df object represents the raw spectral data with minimal preprocessing. Use mia_spectra_grouped_df for the pre-grouped variant.

This portable data.frame format enables export to CSV, Parquet, JSON, and other formats for cross-platform distribution via r-universe. It is automatically converted to a Spectra object by spectra_to_queries().

Source

Data extracted from MIADB, a public spectral library of monoterpene indole alkaloids. Original source: <https://miadb.imsb.qcb.usp.br/>

Accompanying publication: Szwarc, S., Rutz, A., Lee, K. et al. (2025). "Translating community-wide spectral library into actionable chemical knowledge: a proof of concept with monoterpene indole alkaloids." *Journal of Cheminformatics*, 17:62. <https://doi.org/10.1186/s13321-025-01009-0>

Examples

```
## Not run:
# Load the raw spectra
utils::data(mia_spectra_df)
nrow(mia_spectra_df) # 321 spectra

# Convert to Spectra object (for advanced users)
# spectra_obj <- SpectraToQueries::df_to_spectra(mia_spectra_df)

## End(Not run)
```

mia_spectra_grouped_df

Grouped mass spectrometry spectra from MIADB (as portable data.frame)

Description

A data.frame containing 321 grouped mass spectra from the MIADB (Monoterpene Indole Alkaloid Database), combined by spectrum title. This is the primary example dataset used in the spectra_to_queries() function. Stored as a portable data.frame with m/z and intensity peaks as list columns.

Usage

```
mia_spectra_grouped_df
```

Format

A data.frame with 321 rows and 36 columns:

- **Metadata columns:** msLevel, rtime, acquisitionNum, TITLE, PRECURSOR_MZ, SKELETON, and other MS/MS spectrum attributes
- **mz (list):** Fragment m/z values for each spectrum
- **intensity (list):** Fragment intensities for each spectrum

Total of approximately 48,510 peaks across all spectra (median ~132 peaks per spectrum).

Details

This dataset is grouped by spectrum title (via `Spectra::combineSpectra()`) to simplify downstream analysis. It is the recommended example for demonstrating the `spectra_to_queries()` workflow.

This portable `data.frame` format enables export to CSV, Parquet, JSON, and other formats for cross-platform distribution via `r-universe`. It is automatically converted to a `Spectra` object by `spectra_to_queries()`.

Source

Data extracted and pre-grouped from MIADB, a public spectral library of monoterpene indole alkaloids. Original source: <https://miadb.imsb.qcb.usp.br/>

Accompanying publication: Szwarc, S., Rutz, A., Lee, K. et al. (2025). "Translating community-wide spectral library into actionable chemical knowledge: a proof of concept with monoterpene indole alkaloids." *Journal of Cheminformatics*, 17:62. <https://doi.org/10.1186/s13321-025-01009-0>

Examples

```
## Not run:
# Load the grouped spectra
utils::data(mia_spectra_grouped_df)
nrow(mia_spectra_grouped_df) # 321 spectra
table(mia_spectra_grouped_df$SKELETON) # Skeleton distribution

# Convert to Spectra object (for advanced users)
# spectra_obj <- SpectraToQueries::df_to_spectra(mia_spectra_grouped_df)

## End(Not run)
```

normalize_peaks

<i>Normalize peaks</i>

Description

Normalize peaks

Usage

```
normalize_peaks()
```

Examples

```
NULL
```

`perform_list_of_queries`*Perform list of queries*

Description

Perform list of queries

Usage

```
perform_list_of_queries(index, ions_list, query_data, dalton, ppm)
```

Arguments

<code>index</code>	Index of the ion list to process
<code>ions_list</code>	List of ions for queries
<code>query_data</code>	Data frame with columns <code>mz</code> , <code>precursor_mz</code> , and <code>skeleton</code>
<code>dalton</code>	Tolerance in Dalton
<code>ppm</code>	Tolerance in parts per million

Value

Data frame with target and value columns

Examples

```
NULL
```

`perform_list_of_queries_progress`*Perform list of queries*

Description

Perform list of queries

Usage

```
perform_list_of_queries_progress(  
  ions_list,  
  spectra,  
  dalton,  
  ppm,  
  show_progress = TRUE  
)
```

Arguments

ions_list	List of ion combinations for queries
spectra	Spectra object to search
dalton	Tolerance in Dalton
ppm	Tolerance in parts per million
show_progress	Logical: show progress bar (default TRUE)

Value

List of query results

Examples

NULL

perform_query	<i>Perform query</i>
---------------	----------------------

Description

Perform query

Usage

```
perform_query(all_mz, precursor_mz, frags, nls, dalton, ppm)
```

Arguments

all_mz	List of numeric vectors of m/z values for each spectrum
precursor_mz	Numeric vector of precursor m/z values for each spectrum
frags	Fragment masses to search for
nls	Neutral losses to search for
dalton	Tolerance in Dalton
ppm	Tolerance in parts per million

Value

Filtered spectra object

Examples

NULL

prepare_query_data	<i>Prepare spectra data for querying</i>
--------------------	--

Description

Extracts m/z, precursor m/z, and skeleton labels once, so perform_query() never has to touch the Spectra backend again.

Usage

```
prepare_query_data(spectra)
```

Arguments

spectra	Spectra object
---------	----------------

Examples

```
NULL
```

spectra_to_queries	<i>Spectra to queries</i>
--------------------	---------------------------

Description

This function converts spectra to diagnostic ion queries.

Usage

```
spectra_to_queries(  
  spectra = NULL,  
  export = "data/interim/queries.tsv",  
  dalton = 0.01,  
  decimals = 4L,  
  intensity_min = 0,  
  ions_max = 10L,  
  n_skel_min = 5L,  
  n_spec_min = 3L,  
  ppm = 30,  
  mcc_min = 0,  
  precision_min = 0,  
  recall_min = 0,  
  zero_val = 0  
)
```

Arguments

spectra	Spectra object or path to MGF file. If NULL, loads the example spectra dataset. If the string "grouped", loads the grouped example spectra. Otherwise, a path to an MGF file or a Spectra object.
export	Export path
dalton	Tolerance in Dalton. Default to 0.01
decimals	Number of decimals for rounding. Default to 4L
intensity_min	Minimal intensity. Default to 0.0
ions_max	Maximal number of ions in the query. Default to 10L
n_skel_min	Minimal number of individuals per skeleton. Default to 5L
n_spec_min	Minimal number of individuals where a signal has to be found. Default to 3L
ppm	Tolerance in parts per million Default to 30.0
mcc_min	Minimal single-ion Matthews Correlation Coefficient. Default to 0.0
precision_min	Minimal single ion precision (pre-filter before MCC). Default to 0.0
recall_min	Minimal single ion recall (pre-filter before MCC). Default to 0.0
zero_val	Zero value for intensity. Default to 0.0

Value

A file with diagnostic query ions

Examples

```
## Not run:
# Run with default spectra
spectra_to_queries()

# Or explicitly load and pass spectra_grouped
data(spectra_grouped)
spectra_to_queries(spectra = spectra_grouped)

## End(Not run)
```

Index

* datasets

- [mia_spectra_df, 8](#)
 - [mia_spectra_grouped_df, 9](#)
- [combine_ions_minimal, 2](#)
- [contains_all_mz, 3](#)
- [contains_all_mz_tol, 3](#)
- [create_dir, 4](#)
- [create_matrix, 4](#)
- [filter_matrix, 5](#)
- [fix_binned_mzs, 5](#)
- [generate_combinations, 6](#)
- [generate_combinations_progress, 7](#)
- [harmonize_mzs, 7](#)
- [mia_spectra_df, 8](#)
- [mia_spectra_grouped_df, 9](#)
- [normalize_peaks, 10](#)
- [perform_list_of_queries, 11](#)
- [perform_list_of_queries_progress, 11](#)
- [perform_query, 12](#)
- [prepare_query_data, 13](#)
- [spectra_to_queries, 13](#)