Package: MsBackendMongoDb (via r-universe)

June 3, 2026

Title NoSQL-based Mass Spectrometry Data Backend

Version 0.97.3

Description Mass spectrometry (MS) data backend for data storage in NoSQL MongoDB dagabases. Objects from this package expand the data storage and handling infrastructure of the Spectra Bioconductor package. This package adds support to store and retrieve MS data from MongoDB databases.

Depends R (>= 4.5.0), Spectra (>= 1.19.8)

Imports jsonlite, mongolite, S4Vectors, MsCoreUtils, ProtGenerics, IRanges, fastmatch, methods, DBI

Suggests testthat, knitr (>= 1.1.0), roxygen2, BiocStyle (>= 2.5.19), msdata, rmarkdown, mzR

License Artistic-2.0

Encoding UTF-8

VignetteBuilder knitr

BugReports https://github.com/RforMassSpectrometry/MsBackendMongoDb/issues

URL https://github.com/RforMassSpectrometry/MsBackendMongoDb

biocViews Infrastructure, MassSpectrometry, Metabolomics, DataImport, Proteomics

Roxygen list(markdown=TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libuv1-dev libssl-dev libsasl2-dev

Repository https://rformassspectrometry.r-universe.dev

Date/Publication 2026-01-07 12:08:58 UTC

RemoteUrl https://github.com/rformassspectrometry/msbackendmongodb

RemoteRef HEAD

RemoteSha 8666897aa2e56390b42dc4ba2dd21b71af1693c6

Contents

`connectMsBackendMongoDb` *Create a connection to MongoDb*

Description

Creates connections to required MongoDb collections used by the MsBackendMongoDb.

Usage

```
connectMsBackendMongoDb(
  db = "spectra_db",
  url = "mongodb://localhost",
  clean = FALSE
)
```

Arguments

`db`	`character(1)` scalar, name of the MongoDb database.
`url`	`character(1)` a MongoDb server URL.
`clean`	`logical(1)` whether the collection should be *cleaned*, i.e., if all content should be deleted. Be careful with this parameter as setting to `TRUE` will delete evantually persent content from the database.

Value

A named `list()` of `mongo` connection objects.

Author(s)

Ahlam Mentag

MsBackendMongoDb Spectra *MS backend storing data in a MongoDb database*

Description

MsBackendMongoDb is an implementation of the Spectra::MsBackend() class for Spectra::Spectra() objects, storing and retrieving MS data from a MongoDb database.

This backend allows read-only access to spectra and peaks stored in MongoDb, but supports temporary modifications of spectra variables via caching.

Usage

```
## S4 method for signature 'MsBackendMongoDb'
show(object)

## S4 method for signature 'MsBackendMongoDb'
backendInitialize(object, dbcon, data, ...)

## S4 method for signature 'MsBackendMongoDb'
dataStorage(object)

## S4 method for signature 'MsBackendMongoDb'
peaksData(object, columns = c("mz", "intensity"))

## S4 method for signature 'MsBackendMongoDb'
peaksVariables(object)

## S4 method for signature 'MsBackendMongoDb'
mz(object)

## S4 replacement method for signature 'MsBackendMongoDb'
mz(object) <- value

## S4 method for signature 'MsBackendMongoDb'
intensity(object)

## S4 replacement method for signature 'MsBackendMongoDb'
intensity(object) <- value

## S4 method for signature 'MsBackendMongoDb'
spectraNames(object)

## S4 replacement method for signature 'MsBackendMongoDb'
spectraNames(object) <- value

## S4 method for signature 'MsBackendMongoDb'
```

```
spectraData(object, columns = spectraVariables(object))

## S4 method for signature 'MsBackendMongoDb'
x[i, j, ..., drop = FALSE]

## S4 method for signature 'MsBackendMongoDb,ANY'
extractByIndex(object, i)

## S4 method for signature 'MsBackendMongoDb'
reset(object)

## S4 method for signature 'MsBackendMongoDb'
supportsSetBackend(object, ...)
```

Arguments

object	A MsBackendMongoDb instance
dbcon	list() of MongoDb collections, e.g., list(compounds_info_coll = mongo_conn)
data	Optional DataFrame with data to insert into MongoDb
...	Additional arguments passed internally
columns	Character() vector of columns to fetch for peaksData or spectraData
value	Replacement values for setter functions
x	for [i]: a MsBackendMongoDb instance
i	integer() Indices to subset the backend
j	integer() Ignored.
drop	logical(1), ignored for subsetting

Value

Depends on the method. See individual function documentation.

Creation of backend objects

New backend objects can be created using:

```
con <- connectMsBackendMongoDb(db = <database name>, url = <db url>)
backend <- backendInitialize(MsBackendMongoDb(), dbcon = con)
```

- dbcon: A list of MongoDb collections (e.g. from mongolite::mongo()) as created by the connectMsBackendMongoDb().
- backendInitialize(): populates the object with spectrum IDs and caches metadata. If data is provided, it can insert data into the collection.

Accessing data

- `peaksData(object, columns = c("mz", "intensity"))`: returns a `list` of matrices containing the peak data for each spectrum.
- `peaksVariables(object)`: returns available columns for `peaksData`.
- `intensity(object)`, `mz(object)`: return `SimpleList` objects with numeric intensity or m/z values for each spectrum.
- `spectraData(object, columns)`: returns a `DataFrame` with requested spectrum metadata columns.
- `spectraNames(object)`: returns `spectrum_id` values as character.

Read-only data

- `intensity<-` and `mz<-` are **not supported**.
- `spectraNames<-` is **not supported**.

Subsetting and extraction

- `[i]` and `extractByIndex(object, i)` subset the backend by spectrum indices. Original data in the database is never changed.
- `reset(object)` restores the backend to its original state, removing subsetting and cached variables.

Miscellaneous

- `dataStorage(object)` returns a string describing the MongoDb collections used by the backend.

Implementation notes

Internally, the backend stores:

- `spectraIds`: integer vector of spectrum IDs
- `dbcon`: MongoDb connection(s)
- `.collections`: list of collection names
- `peak_fun`: function used to fetch peak data
- `localData`: cached spectrum variables
- `nspectra`: total number of spectra

Data access functions use `peak_fun` to fetch peaks on demand.

Note

`MsBackendMongoDb` keeps a connection to a MongoDb collection(s) and fetches spectra and peaks data on demand. By storing only the primary keys (`spectrum_id`) in memory, it ensures minimal memory usage, while peaks and metadata are fetched dynamically.

The backend inherits from `Spectra::MsBackendCached()` and supports temporary modification of spectra variables via the `$<-` operator. Original data in the database cannot be changed (intensities and m/z values are read-only).

Author(s)

Ahlam Mentag

Index