Package: MsExperimentStash (via r-universe)

November 25, 2026

Title Save and restore MsExperiment objects in interoperable file formats

Version 1.0.0

Description The serialization mechanism of R allows to save and load R data objects in a binary format, that can however not be read by other programming languages or software. The MsExperimentStash package implements the serialization methods from the MsStash package for MsExperiment objects. Data can be stashed in a variety of file formats and types, including HDF5 and JSON-based formats defined by the Bioconductor *alabaster* package. The package provides also functionality to read a full MS experiment, including MS data and experimental metadata information, from the MetaboLights repository.

Depends R (>= 4.5.0), MsStash

Imports methods, alabaster.base, MsExperiment, S4Vectors, Spectra, MsCoreUtils, ProtGenerics, jsonlite

Suggests BiocStyle, testthat, pkgload, callr, fs, alabaster.matrix, alabaster.se, SummarizedExperiment, SpectraStash, MsDataHub, QFeatures, mzR, knitr, MsBackendMetaboLights, MsBackendMetabolomicsWorkbench

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

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

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

biocViews Infrastructure, MassSpectrometry, Metabolomics, DataImport, Proteomics

Roxygen list(markdown=TRUE)

Config/roxygen2/version 8.1.1

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 23:28:09 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 7683274765ec7061df37f06c5eb6b018f3d0ee5a

Contents

MetaboLightsParam 2
MsExperimentFilesStash 4
MsExperimentStash 6
MwbParam 9

Index **12**

`MetaboLightsParam` *Load Content from a MetaboLights Study*

Description

The `MetaboLightsParam` class and the associated `readMsObject()` method allow users to load an MsExperiment::MsExperiment object from a study in the MetaboLights database (https://www.ebi.ac.uk/metabolights/index) by providing its unique study identifier (parameter `mtblsId`). This function is particularly useful for directly importing metabolomics data (MS data and experimental metadata) into an `MsExperiment` object for further analysis in the R environment.

It is important to note that at present it is only possible to *read* (import) data from MetaboLights, but not to *save* data to MetaboLights.

If the study contains multiple assays (e.g. measurements performed in positive or negative polarity or data subsets with different liquid chromatography setups used), the user will be prompted to select which assay to load. The resulting `MsExperiment` object will include a `sampleData` slot populated with data extracted from the selected assay.

Users can define how to filter this `sampleData` table by specifying a few parameters. The `keepOntology` parameter is set to `TRUE` by default, meaning that all ontology-related columns are retained. If set to `FALSE`, they are removed. If ontology columns are kept, some column names may be duplicated and therefore numbered. The order of these columns is important, as it reflects the assay and sample information available in MetaboLights.

The `keepProtocol` parameter is also set to `TRUE` by default, meaning that all columns related to protocols are kept. If set to `FALSE`, they are removed. The `simplify` parameter (default `simplify = TRUE`) allows to define whether duplicated columns or columns containing only missing values should be removed. In the case of duplicated content, only the first occurring column will be retained.

Further filtering can be performed using the `filePattern` parameter of the `MetaboLightsParam` object. The default for this parameter is `"mzML$|CDF$|cdf$|mzXML$"`, which corresponds to the supported raw data file types.

Usage

```
MetaboLightsParam(
  mtblsId = character(),
  assayName = character(),
  filePattern = "mzML$|CDF$|cdf$|mzXML$"
)

## S4 method for signature 'MsExperiment,MetaboLightsParam'
readMsObject(
  object,
  param,
  keepOntology = TRUE,
  keepProtocol = TRUE,
  simplify = TRUE,
  ...
)
```

Arguments

`mtblsId`	`character(1)` The MetaboLights study ID, which should start with "MTBL". This identifier uniquely specifies the study within the MetaboLights database.
`assayName`	`character(1)` The name of the assay to load. If the study contains multiple assays and this parameter is not specified, the user will be prompted to select which assay to load.
`filePattern`	`character(1)` A regular expression pattern to filter the raw data files associated with the selected assay. The default value is `"mzML$\|CDF$|cdf$\|mzXML$"`, corresponding to the supported raw data file types.
`object`	For `readMsObject()`: a `MsExperiment` instance.
`param`	For `readMsObject()`: a `MetaboLightsParam` object.
`keepOntology`	`logical(1)` Whether to keep columns related to ontology in the object's `sampleData()`. Default is `TRUE`.
`keepProtocol`	`logical(1)` Whether to keep columns related to protocols information in the object's `sampleData()`. Default is `TRUE`.
`simplify`	`logical(1)` Whether to simplify the `sampleData()` table by removing columns filled with NAs or duplicated content. Default is `TRUE`.
`...`	Currently ignored.

Value

`readMsObject()` returns an `MsExperiment` object with the `sampleData()` populated with MetaboLights sample and assay information and the experiment's MS data loaded as a Spectra::Spectra object.

Author(s)

Philippine Louail

See Also

- MsExperiment::MsExperiment object.
- MsBackendMetaboLights::MsBackendMetaboLights for retrieving MS data files from MetaboLights.
- MetaboLights for accessing the MetaboLights database.

Examples

```
library(MsExperiment)
## Load a study with the mtblsId "MTBLS39" and selecting specific file
## pattern as well as removing ontology and protocol information in the
## metadata.
param <- MetaboLightsParam(mtblsId = "MTBLS39", filePattern = "63A.cdf")
ms_experiment <- readMsObject(MsExperiment(), param, keepOntology = FALSE,
                              keepProtocol = FALSE)
ms_experiment

## The object's sampleData contains information loaded from MetaboLights
sampleData(ms_experiment)

## The MS data files were downloaded and cached; the data is available
## through the object's `Spectra`
spectra(ms_experiment)
```

MsExperimentFilesStash

Stash for MsExperimentFiles

Description

The MsExperiment::MsExperimentFiles class stores files (or rather file names) that are part of a mass spectrometry experiment.

The supported stash formats for MsExperimentFiles objects are listed in the sections below.

Usage

```
## S4 method for signature 'MsExperimentFiles,PlainTextParam'
saveMsObject(object, param, ...)

## S4 method for signature 'MsExperimentFiles,PlainTextParam'
readMsObject(object, param, ...)

## S4 method for signature 'MsExperimentFiles'
saveObject(x, path, ...)

## S4 method for signature 'MsExperimentFiles,AlabasterParam'
saveMsObject(object, param, ...)
```

```
## S4 method for signature 'MsExperimentFiles,AlabasterParam'
readMsObject(object, param, ...)
```

Arguments

object	An MsExperimentFiles object.
param	An MsStash::AlabasterParam or MsStash::PlainTextParam.
...	Currently ignored.
x	An MsExperimentFiles object.
path	For saveObject(): character(1) with the path where the object should be stored into.

Value

readMsObject() returns a MsExperiment::MsExperimentFiles object.

alabaster-based format, AlabasterParam

The MsExperimentFiles stash folder contains the alabaster-specific *OBJECT* file and a sub-folder *x* with the MsExperimentFiles content serialized by *alabaster.base*.

Text-file format, PlainTextParam

The text-file format stash folder for MsExperimentFile objects contains a file *ms_experiment_files.txt* with two tabulator separated columns *name* and *files*. Each row (except the first) is one element of the MsExperimentFile, the first defining the object's names and the second its content, which represents a characted vector with the file name(s), separated by a "|" (if more than one).

Author(s)

Johannes Rainer

Examples

```
library(MsExperiment)

fls <- MsExperimentFiles(list(input = c("file.mzML", "file2.mgf")))

## Define the path to the stash
d <- file.path(tempdir(), "ms_file_stash")

## Stash the object in alabaster format
saveMsObject(fls, AlabasterParam(d))

## The content of the stash: subfolder x contains the *character list*
## saved through the *alabaster.base* package.
library(fs)
dir_tree(d)
```

```
## Restore the object from stash
res <- readMsObject(MsExperimentFiles(), AlabasterParam(d))
res

## In addition, it is possible to read the object also with the
## *alabaster.base* functionality
library(alabaster.base)
res <- readObject(d)
```

MsExperimentStash MsExperiment *Stash*

Description

MsExperiment objects can be stored to (or read from) *MsExperimentStash*es using the saveMsObject() and readMsObject() functions which take a second argument parameter to select and configure the format of the stash.

The supported stash formats are listed in the sections below.

Usage

```
## S4 method for signature 'MsExperiment,PlainTextParam'
saveMsObject(object, param, ...)

## S4 method for signature 'MsExperiment,PlainTextParam'
readMsObject(object, param, ...)

## S4 method for signature 'MsExperiment'
saveObject(x, path, ...)

## S4 method for signature 'MsExperiment,AlabasterParam'
saveMsObject(object, param, ...)

## S4 method for signature 'MsExperiment,AlabasterParam'
readMsObject(object, param, ...)
```

Arguments

object	A MsExperiment object.
param	The parameter object to select and configure the stash format. Either MsStash::AlabasterParam or MsStash::PlainTextParam.
...	For saveMsObject(): optional arguments passed down to the saveMsObject() function to stash the Spectra object (if present), such as consolidate. For readMsObject(): optional arguments for the readMsObject() call to restore the Spectra object (such as spectraPath). See SpectraStash::SpectraStash for more information.
x	A MsExperiment object.
path	For saveObject(): character(1) with the path to the folder for the stash.

Value

`readMsObject()` and `readObject()` return an `MsExperiment` object with content read from the specified MsExperimentStash.

alabaster-based format, `AlabasterParam`

This stash format is the most complete and reliable way for long-term (and portable) storage of an `MsExperiment`. Objects can be saved or read from this stash format either using the `saveMsObject()` and `readMsObject()` functions or also using the `alabaster.base::saveObject()` and `alabaster.base::readObject()` functions. Data from the object's slots are stored to their respective folders (using alabaster functionality). These folders are:

- *experiment_files*: the content of the `@experimentFiles` slot, stored as a MsExperimentFilesStash.
- *metadata*: the content of the object's `@metadata` slot.
- *other_data*: the content of the object's `@otherData` slot. Note that export fails if object types are stored in this slot without available *alabaster* export functionality.
- *qdata*: the content of the object's `@qdata` slot (if present). Currently only SummarizedExperiment objects are supported.
- *sample_data*: the object's `sampleData` data frame.
- *sample_data_links*: the content of the object's `@sampleDataLinks` slot defining the mapping between rows in `sampleData` and other entities in the object, such as e.g. spectra.
- *sample_data_links_mcols*: the metadata content of the `@sampleDataLinks`.
- *spectra*: the Spectra::Spectra object with the MS data (if present). The respective SpectraStash::SpectraStash functionality is used to export this data. Note that not all `MsBackend` types might be supported. In that case, the backend should be switched to one of the `MsBackend`s from the *Spectra* package using the `Spectra::setBackend()` function.

Text-file format, `PlainTextParam`

`MsExperiment` objects can also be saved to plain text files as a MsExperimentStash in text file format. Note however that currently only some of the object's content can be saved in that format. Content of slots `@qdata`, `@metadata` and `@otherData` (if present) are **not** stored to the stash. The text file-based stash directory contains the following files:

- *ms_experiment_sample_data.txt*: tabulator delimited text file with the content of the `MsExperiment`'s `sampleData()`. This file is always saved.
- *ms_experiment_sample_data_links_.txt*: a two column tab-delimited text file with the mapping between rows in `sampleData()` and elements in other slots of the `MsExperiment`. The name of the respective slot is used as file name suffix. This file is only generated when *sample data links* are present.
- *ms_experiment_link_mcols.txt*: tab-delimited text file with the metadata content of the `@sampleDataLink` slot. This file is only generated when *sample data links* are present.
- *ms_experiment_files.txt*: the object's MsExperiment::MsExperimentFiles. See MsExperimentFilesStash for information on the format. This file is only generated if `experimentFiles()` are present in the object.

If the MsExperiment contained a Spectra::Spectra object, it is saved to the main *MsExperimentStash* folder. A different set of files might be stored, depending on the MsBackend used. This can also include raw MS data files if parameter consolidate = TRUE is used in saveMsObject(). See SpectraStash::SpectraStash for more information.

Retrieve MS data from *MetaboLights*

In addition to the MsExperimentStash formats for storage of MsExperiment objects, it is possible to load data from a metabolomics study directly from the MetaboLights repository. See MetaboLightsParam for more information.

Retrieve MS data from *Metabolomics Workbench*

In addition to the MsExperimentStash formats for storage of MsExperiment objects, it is possible to load data from a metabolomics study directly from the Metabolomics Workbench repository. See MwbParam for more information.

Note

Overwriting an existing *MsExperimentStash* is not allowed.

Serializing MsExperiment objects containing a QFeatures object is currently not supported.

Author(s)

Philippine Louail, Johannes Rainer

Philippine Louail

See Also

MetaboLightsParam for loading an MsExperiment from the MetaboLights public repository.

MwbParam for loading an MsExperiment from the Metabolomics Workbench public repository.

Examples

```
## Example MS data files
library(Spectra)
library(MsExperiment)
library(MsDataHub)
fls <- c(X20171016_POOL_POS_1_105.134.mzML(),
    X20171016_POOL_POS_3_105.134.mzML())

## Create a MsExperiment for the two example files
mse <- readMsExperiment(fls, data.frame(name = c("A", "B"), index = 1:2))

## Define the path where to create the MsExperimentStash
d <- file.path(tempdir(), "ms_experiment_stash")

## Save the MsExperiment to a stash in alabaster format; Note: with
## `consolidate = TRUE` the MS data files are also copied into the
## stash
```

```
saveMsObject(mse, AlabasterParam(d), consolidate = TRUE)

## Show the content of the stash folder
library(fs)
dir_tree(d)

## Restore the object from the stash
res <- readMsObject(MsExperiment(), AlabasterParam(d))
res

sampleData(res)

spectra(res)

## Save the MsExperiment to a stash in plain text format.
d <- file.path(tempdir(), "ms_experiment_text_stash")
saveMsObject(mse, PlainTextParam(d))

## Show the content of the stash folder
dir_tree(d)

## Restore the object
res <- readMsObject(MsExperiment(), PlainTextParam(d))
res
```

`MwbParam` *Load Content from a Metabolomics Workbench Study*

Description

The `MwbParam` class and the associated `readMsObject()` method allow users to load an MsExperiment::MsExperiment object from a study in the Metabolomics Workbench database (https://metabolomicsworkbench.org/) by providing its unique study identifier (parameter `mwbId`). This function is particularly useful for directly importing metabolomics data (MS data and experimental metadata) into an `MsExperiment` object for further analysis in the R environment.

It is important to note that at present it is only possible to *read* (import) data from Metabolomics Workbench, but not to *save* data to Metabolomics Workbench.

If the study contains multiple analyses, the user will be prompted to select which analysis to load. The resulting `MsExperiment` object will include a `sampleData` slot populated with sample metadata from the selected analysis.

Further filtering can be performed using the `filePattern` parameter of the `MwbParam` object. The default for this parameter is `"mzML$|CDF$|cdf$|mzXML$"`, which corresponds to the supported raw data file types.

Usage

```
MwbParam(
  mwbId = character(),
```

```
  analysisId = character(),
  filePattern = "mzML$|CDF$|cdf$|mzXML$"
)

## S4 method for signature 'MsExperiment,MwbParam'
readMsObject(object, param, ...)
```

Arguments

mwbId	character(1) The Metabolomics Workbench study ID, which should start with "ST". This identifier uniquely specifies the study within the Metabolomics Workbench database.
analysisId	character(1) The Metabolomics Workbench analysis ID, which should start with "AN". This identifier uniquely specifies the analysis within the Metabolomics Workbench database.
filePattern	character(1) A regular expression pattern to filter the raw data files associated with the selected assay. The default value is "mzML$\|CDF$|cdf$\|mzXML$", corresponding to the supported raw data file types.
object	For readMsObject(): a MsExperiment instance.
param	For readMsObject(): a MwbParam object.
...	Currently ignored.

Value

readMsObject() returns an MsExperiment object with the sampleData() populated with Metabolomics Workbench sample information and the experiment's MS data loaded as a Spectra::Spectra object.

Author(s)

Gabriele Tomè

See Also

- MsExperiment::MsExperiment object.
- MsBackendMetabolomicsWorkbench::MsBackendMetabolomicsWorkbench for retrieving MS data files from Metabolomics Workbench
- Metabolomics Workbench for accessing the Metabolomics Workbench database.

Examples

```
library(MsExperiment)
## Load a study with the mwbId "ST002115" and selecting specific file
## pattern as well as removing ontology and protocol information in the
## metadata.
param <- MwbParam(mwbId = "ST002115", analysisId = "AN003513",
                  filePattern = "01_RP.mzXML")
ms_experiment <- readMsObject(MsExperiment(), param)
ms_experiment
```

```
## The object's sampleData contains information loaded from Metabolomics
## Workbench
sampleData(ms_experiment)

## The MS data files were downloaded and cached; the data is available
## through the object's `Spectra`
spectra(ms_experiment)
```

Index

alabaster.base::readObject(), *7*
alabaster.base::saveObject(), *7*

MetaboLightsParam, 2, *8*
MsBackendMetaboLights::MsBackendMetaboLights, *4*
MsBackendMetabolomicsWorkbench::MsBackendMetabolomicsWorkbench, *10*
MsExperiment::MsExperiment, *2*, *4*, *9*, *10*
MsExperiment::MsExperimentFiles, *4*, *5*, *7*
MsExperimentFilesStash, 4, *7*
MsExperimentStash, 6
MsStash::AlabasterParam, *5*, *6*
MsStash::PlainTextParam, *5*, *6*
MwbParam, *8*, 9

readMsObject,MsExperiment,AlabasterParam-method *(*MsExperimentStash*)*, 6
readMsObject,MsExperiment,MetaboLightsParam-method *(*MetaboLightsParam*)*, 2
readMsObject,MsExperiment,MwbParam-method *(*MwbParam*)*, 9
readMsObject,MsExperiment,PlainTextParam-method *(*MsExperimentStash*)*, 6
readMsObject,MsExperimentFiles,AlabasterParam-method *(*MsExperimentFilesStash*)*, 4
readMsObject,MsExperimentFiles,PlainTextParam-method *(*MsExperimentFilesStash*)*, 4

saveMsObject,MsExperiment,AlabasterParam-method *(*MsExperimentStash*)*, 6
saveMsObject,MsExperiment,PlainTextParam-method *(*MsExperimentStash*)*, 6
saveMsObject,MsExperimentFiles,AlabasterParam-method *(*MsExperimentFilesStash*)*, 4
saveMsObject,MsExperimentFiles,PlainTextParam-method *(*MsExperimentFilesStash*)*, 4
saveObject,MsExperiment-method *(*MsExperimentStash*)*, 6
saveObject,MsExperimentFiles-method *(*MsExperimentFilesStash*)*, 4
Spectra::setBackend(), *7*
Spectra::Spectra, *3*, *7*, *8*, *10*
SpectraStash::SpectraStash, *6–8*
SummarizedExperiment, *7*