Package: MsBackendMetabolomicsWorkbench (via r-universe)

June 9, 2026

Title Retrieve Mass Spectrometry Data from Metabolomics Workbench

Version 0.1.4

Description Metabolomics Workbench is one of the main public repositories for storage of metabolomics experiments. The MsBackendMetabolomicsWorkbench package provides functionality to retrieve and represent mass spectrometry (MS) data from Metabolomics Workbench. Data files are downloaded and cached locally avoiding repetitive downloads. MS data from metabolomics experiments can thus be directly and seamlessly integrated into R-based analysis workflows with the Spectra and MsBackendMetabolomicsWorkbench package.

Depends R (>= 4.2.0), Spectra (>= 1.15.12)

Imports rvest, httr2, curl, ProtGenerics, BiocFileCache, S4Vectors, methods, progress, stringr, MsCoreUtils (>= 1.23.9), jsonlite, plyr, tidyr, archive

Suggests testthat, rmarkdown, mzR, knitr, BiocStyle

License Artistic-2.0

Encoding UTF-8

VignetteBuilder knitr

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

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

biocViews Infrastructure, MassSpectrometry, Metabolomics, DataImport, Proteomics

Roxygen list(markdown=TRUE)

RoxygenNote 8.0.0

Config/pak/sysreqs cmake make libarchive-dev libicu-dev libuv1-dev libxml2-dev libssl-dev

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

Date/Publication 2026-06-09 11:24:42 UTC

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

RemoteRef HEAD

RemoteSha 619697c0e5a3a5de04ae3bf8522687a429808c07

Contents

`MetabolomicsWorkbench-utils`

Utility functions for the Metabolomics Workbench repository

Description

Utility functions to interact with the Metabolomics Workbench (MWB) repository, including listing, downloading, caching, and querying data files and study metadata.

- `mwb_cached_data_files()`: lists locally cached data files from Metabolomics Workbench. Since this function evaluates only local content it does not require an internet connection. With the default parameters all available data files are listed. The parameters can be used to restrict the lookup.
- `mwb_list_files()`: returns the available files for the specified Metabolomics Workbench data set by submitting a POST request to the Metabolomics Workbench archive contents endpoint. The function returns a `data.frame` with columns `"zip_file"` and `"sample_file"` containing the archive name and the file name within that archive. Parameter `pattern` allows to filter the results by matching against the `"sample_file"` column. This function requires an active internet connection.
- `mwb_rest_request()`: queries the Metabolomics Workbench REST API for a given study/analysis ID and output item (e.g. `"summary"`, `"factors"`). Returns the raw response as a `character` string in the format specified by `outputFormat` (`"json"` or `"txt"`). This function requires an active internet connection.
- `mwb_ftp_list_files()`: queries the Metabolomics Workbench FTP server for a given experiment ID and returns the related files. Parameter `pattern` allows to filter the results. In contrast to `mwb_list_files()`, this function lists only the files on the FTP server (like the zip file of the experiment), while `mwb_list_files()` lists the files contained within the zip file. Other files may also be present on the FTP server. This function requires an active internet connection.
- `mwb_ftp_download()`: download files from Metabolomics Workbench FTP server for a given experiment ID. Use `pattern` to filter files by name using a regular expression (by default all files are downloaded). Use `path` to set the destination directory for downloaded files. Only files listed by `mwb_ftp_list_files()` can be downloaded.

- `mwb_metadata()`: retrieves the metadata of a given MWB data set as a `list` with two `data.frame`: one with the metadata of the experiment and one with the sample annotation. The function handles the case of multiple analysis IDs by combining the metadata of all analysis IDs into a single `data.frame` for the experiment and a single `data.frame` for the sample annotation. This function requires an active internet connection.
- `mwb_sync_data_files()`: synchronize data files of a specified MWB data set eventually downloading and locally caching them. Parameter `fileName` allows to specify names of selected data files to sync.
- `mwb_delete_cache()`: removes all local content for the mwb data set with ID `mwbId`. This will delete eventually present locally cached data files for the specified data set. This does not change any other data eventually present in the local `BiocFileCache`.

Usage

```
mwb_list_files(x = character(), pattern = NULL)

mwb_rest_request(
  id = character(),
  idType = c("study_id", "analysis_id"),
  outputItem = character(),
  outputFormat = c("json", "txt")
)

mwb_ftp_list_files(mwbId = character(), pattern = "*")

mwb_ftp_download(
  mwbId = character(),
  pattern = "*",
  path = "./",
  overwrite = FALSE
)

mwb_metadata(mwbId = character())

mwb_sync_data_files(
  mwbId = character(),
  pattern = "mzML$|mzml$|CDF$|cdf$|mzXML$",
  fileName = character(),
  ftp_zip = FALSE
)

mwb_cached_data_files(
  mwbId = character(),
  pattern = "*",
  fileName = character()
)

mwb_delete_cache(mwbId = character())
```

Arguments

x	character(1) with the ID of the MBW data set (usually starting with a *ST* followed by a number).
pattern	for mwb_list_files(), mwb_sync_data_files(), mwb_cached_data_files(), mwb_ftp_list_files and mwb_ftp_download: character(1) defining a pattern to filter the file names, such as pattern = "mzML$" to retrieve the file names of all files of the data set (i.e., files with extension "mzML"). This parameter is passed to the grepl() function.
id	character(1) with the ID of a single Metabolomics Workbench data set/experiment.
idType	for mwb_rest_request(): character(1) defining the type of the ID provided in id. The accepted ID types are "study_id" and "analysis_id". The default is "study_id".
outputItem	for mwb_rest_request(): character(1) defining the metadata to retrieve from Metabolomics Workbench. To get more information about the possible output visit the webpage MBW REST API.
outputFormat	for mwb_rest_request(): character(1) defining the output format of the metadata. The supported output are json and txt.
mwbId	character(1) with the ID of a single Metabolomics Workbench data set/experiment.
path	for mwb_ftp_download(): optional character defining the directory where download the files.
overwrite	for mwb_ftp_download(): logical(1) whether existing files should be overwritten. Defaults to FALSE, in which case files that already exist in path are skipped.
fileName	for mwb_sync_data_files() and mwb_cached_data_files(): optional character defining the names of specific data files of a data set that should be downloaded and cached.
ftp_zip	for mwb_sync_data_files(): logical(1) download the complete zip of the experiment from the FTP server. Defaults to FALSE, in which case the files are downloaded singularly via POST request.

Details

Metabolomics Workbench provides metadata through a REST API. MS data files can be obtained in two ways:

1. Downloading the full *zip* archive from the FTP server. A POST request to the MWB archive page gets the correct *zip* archive name for a MWB ID. The archive contains all files of the experiment, which may include also unneeded files. If only a subset of files is needed, the second option is more efficient.
2. Download individual files using a two-step POST-based procedure: query the MWB archive page to get exact file names. Then, download each file via POST request.

Value

- For mwb_list_files(): data.frame with columns zip_file and sample_file containing, respectively, the archive name and the relative file within that archive

- For `mwb_rest_request()`: `character(1)` with the raw REST API response body, formatted as JSON or plain text depending on `outputFormat`.
- For `mwb_sync_data_files()` and `mwb_cached_data_files()`: a `data.frame` with the MWB ID, the name(s) and remote and local file names of the synchronized data files.
- For `mwb_ftp_list_files`: `character` with the files in FTP server for a specific ID.
- For `mwb_metadata`: `list` with two `data.frame`: one with the metadata of the experiment and one with the sample annotation.

Author(s)

Gabriele Tomè, Johannes Rainer, Philippine Louail

Examples

```
## Retrieve available files for the data set ST002115
mwb_list_files("ST002115")

## Retrieve the available .mzML files.
A1_mzMLfiles <- mwb_list_files("ST000016", pattern = "A1")
A1_mzMLfiles

## Query the REST API for a study summary in JSON format
mwb_rest_request("ST002115", outputItem = "summary")

## List zip file of the data set ST002115
mwb_ftp_list_files("ST002115")

## Download the file with: `mwb_ftp_download("ST002115", path = tempdir())`
```

`MsBackendMetabolomicsWorkbench` *MsBackend representing MS data from Metabolomics Workbench*

Description

`MsBackendMetabolomicsWorkbench` retrieves and represents mass spectrometry (MS) data from metabolomics studies stored in the Metabolomics Workbench repository, a data resource developed by the NIH Common Fund's Data Repository and Coordinating Center (DRCC) at the San Diego Supercomputer Center, University of California San Diego. The repository provides access to study metadata, processed experimental results, metabolite structures, and reference compound information through a RESTful HTTP API / FTP server / POST request. The backend directly extends the Spectra::MsBackendMzR backend from the *Spectra* package and hence supports MS data in mzML, CDF, and mzXML format. Data in other formats cannot be loaded with `MsBackendMetabolomicsWorkbench`. Upon initialization with the `backendInitialize()` method, the `MsBackendMetabolomicsWorkbench` backend fetches and caches study data files locally using Bioconductor's *BiocFileCache* package, avoiding repeated queries to the remote repository. See the help and vignettes of that package for details on cached data resources. Additional utility functions for management of cached files are also provided by *MsBackendMetabolomicsWorkbench*. See help for `mwb_cached_data_files()` for more information.

Usage

```
MsBackendMetabolomicsWorkbench()

## S4 method for signature 'MsBackendMetabolomicsWorkbench'
backendInitialize(
  object,
  mwbId = character(),
  filePattern = "mzML$|CDF$|cdf$|mzXML$",
  ftp_zip = FALSE,
  offline = FALSE,
  ...
)

## S4 method for signature 'MsBackendMetabolomicsWorkbench'
backendRequiredSpectraVariables(object, ...)

mwb_sync(x, offline = FALSE)
```

Arguments

`object`	an instance of `MsBackendMetabolomicsWorkbench`.
`mwbId`	`character(1)` with the ID of a single MetabolomicsWorkbench data set/experiment.
`filePattern`	`character` with the pattern defining the supported (or requested) file types. Defaults to `filePattern = "mzML$\|CDF$|cdf$\|mzXML$"` hence restricting to mzML, CDF and mzXML files which are supported by *Spectra*'s `MsBackendMzR` backend.
`ftp_zip`	for `mwb_sync_data_files()`: `logical(1)` download the complete zip of the experiment from the FTP server. Defaults to `FALSE`, in which case the files are downloaded singularly via POST request.
`offline`	`logical(1)` whether only locally cached content should be evaluated/loaded.
`...`	additional parameters; currently ignored.
`x`	an instance of `MsBackendMetabolomicsWorkbench`.

Details

The backend uses the BiocFileCache package for caching of the data files. These are stored in the default local *BiocFileCache* cache along with additional metadata that includes the Metabolomics Workbench ID. Note that at present only MS data files in *mzML*, *CDF* and *mzXML* format are supported.

The `MsBackendMetabolomicsWorkbench` backend defines and provides additional spectra variables `"mwb_id"`, `"zip_file"` and `"file_name"` that list the MetabolomicsWorkbench ID, the original zip file name and the original data file name on the Metabolomics Workbench ftp server for each individual spectrum. The `"file_name"` can be used for the mapping between the experiment's samples and the individual data files, respective their spectra.

The `MsBackendMetabolomicsWorkbench` backend is considered *read-only* and does thus not support changing *m/z* and intensity values directly.

Value

- For `MsBackendMetabolomicsWorkbench()`: an instance of `MsBackendMetabolomicsWorkbench`.
- For `backendInitialize()`: an instance of `MsBackendMetabolomicsWorkbench` with the MS data of the specified MetabolomicsWorkbench data set.
- For `backendRequiredSpectraVariables()`: `character` with spectra variables that are needed for the backend to provide the MS data.
- For `mwb_sync()`: the input `MsBackendMetabolomicsWorkbench` with the paths to the locally cached data files being eventually updated.

Initialization and loading of data

New instances of the class can be created with the `MsBackendMetabolomicsWorkbench()` function. Data is loaded and initialized using the `backendInitialize()` function, which accepts parameters `mwbId`, `filePattern` and `ftp_zip`. `mwbId` must be the accession of a **single** existing Metabolomics Workbench study (e.g. `"ST000016"`). Optional parameter `filePattern` defines the pattern used to filter the file names of the MS data files and defaults to data files with file endings of supported MS data formats. Optional parameter `ftp_zip = TRUE` will download the complete zip file of the experiment from the FTP server and extract the data files locally, which can be faster than downloading the files individually via POST request. However if only a subset of the data files is required, it is more efficient to download the files separately *via* POST request with `ftp_zip = FALSE` and `filePattern` set to the desired file name pattern. `backendInitialize()` requires an active internet connection, as the function queries the Metabolomics Workbench via POST request and compares remote file content against locally cached files before synchronizing any changes or updates. This behavior can be bypassed with `offline = TRUE`, in which case only locally cached content is used.

The `backendRequiredSpectraVariables()` function returns the names of the spectra variables required for the backend to provide the MS data.

The `mwb_sync()` function can be used to *synchronize* the local data cache and ensure that all study data files are locally available. The function checks the local cache and downloads any missing data files from the Metabolomics Workbench repository.

Note

To account for transient network failures and high server load on the Metabolomics Workbench endpoint, download functions automatically retry failed requests. An error is raised after 5 consecutive failed attempts. Between each attempt, the function waits for a progressively increasing time period (5 seconds between the first and second attempt, 10 seconds between the second and third, and so forth). The sleep time multiplier can be configured via the `"mwb.sleep_mult"` option (defaults to 5). An active internet connection is required for all non-cached operations; use `offline = TRUE` in `backendInitialize()` to suppress remote requests and rely exclusively on the local *BiocFileCache* cache.

Author(s)

Gabriele Tomè, Philippine Louail, Johannes Rainer

Examples

```
library(MsBackendMetabolomicsWorkbench)

## List files of a MetabolomicsWorkbench data set
mwb_list_files("ST002115")

## Initialize a MsBackendMetabolomicsWorkbench representing all MS
## data files of the data set with the ID "ST002115". This will
## download and cache all files and subsequently load and represent
## them in R.

be <- backendInitialize(MsBackendMetabolomicsWorkbench(),
                        "ST002115",
                        filePattern = "DMSO_01_RP.mzXML$")
be

## The `mwb_sync()` function can be used to ensure that all data
## files are available locally. This function will eventually download
## missing data files or update their paths.
be <- mwb_sync(be)
```

Index