Package: MGnifyR (via r-universe)

November 25, 2026

Type Package

Version 2.0.0

Title R interface to EBI MGnify metagenomics resource

Description MGnifyR is an R interface to MGnify, EBI's resource for the analysis of metagenomic, metatranscriptomic and amplicon data. It searches MGnify studies, samples and analyses with MGnify API v2, and fetches their taxonomic and functional profiles together with sample metadata. The data is returned as TreeSummarizedExperiment or MultiAssayExperiment, so that it can be analysed directly with Bioconductor microbiome tools.

biocViews Infrastructure, DataImport, Metagenomics, Microbiome

License Artistic-2.0

Encoding UTF-8

Depends R (>= 4.5.0), TreeSummarizedExperiment

Imports curl, jsonlite, Matrix, MultiAssayExperiment, S4Vectors, stats, tools, utils

Suggests knitr, quarto, testthat

URL https://github.com/EBI-Metagenomics/MGnifyR

BugReports https://github.com/EBI-Metagenomics/MGnifyR/issues

VignetteBuilder quarto

SystemRequirements Quarto command-line tools (https://github.com/quarto-dev/quarto-cli)

Config/roxygen2/version 8.1.0

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:52:13 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 2a5672a52caf4919fa439292d48272fbaa5a4308

Contents

MGnifyR-package 2
deprecated 2
fetchStudies 6
mgnifyLogin 8
searchStudies 9

Index **12**

`MGnifyR-package` `MGnifyR` *Package.*

Description

`MGnifyR` fetches data from the EBI MGnify database into R. Browse studies with `searchStudies` and fetch their taxonomic profiles as `TreeSummarizedExperiment` with `fetchStudies`. See the vignette for an introduction, About MGnify for general MGnify information, and API documentation for details about MGnify API v2.

Author(s)

Maintainer: Tuomas Borman <tuomas.v.borman@utu.fi> (ORCID)

Authors:

- Tuomas Borman <tuomas.v.borman@utu.fi> (ORCID)
- Ben Allen <ben.allen@ncl.ac.uk>
- Leo Lahti <leo.lahti@iki.fi> (ORCID)

See Also

`TreeSummarizedExperiment`

`deprecated` *Deprecated and defunct functions*

Description

Deprecated and defunct functions

Usage

```
MgnifyClient(
  username = NULL,
  password = NULL,
  useCache = FALSE,
  cacheDir = tempdir(),
  ...
)

getResult(
  x,
  accession,
  get.taxa = "SSU",
  get.func = TRUE,
  output = "TreeSE",
  ...
)

searchAnalysis(x, type, accession, ...)

getMetadata(x, accession, ...)

doQuery(
  x,
  type = "studies",
  accession = NULL,
  as.df = TRUE,
  max.hits = 200,
  ...
)

getData(x, type, accession.type = NULL, accession = NULL, as.df = TRUE, ...)

searchFile(
  x,
  accession,
  type = c("studies", "samples", "analyses", "assemblies", "genomes", "run"),
  ...
)

getFile(x, url, file = NULL, read.func = NULL, ...)

databaseUrl(...)

authTok(...)

useCache(...)
```

```
cacheDir(...)

showWarnings(...)

clearCache(...)

verbose(...)

databaseUrl(x) <- value

authTok(x) <- value

useCache(x) <- value

cacheDir(x) <- value

showWarnings(x) <- value

clearCache(x) <- value

verbose(x) <- value

mgnify_client(...)

mgnify_query(...)

mgnify_analyses_from_samples(...)

mgnify_analyses_from_studies(...)

mgnify_get_download_urls(...)

mgnify_download(...)

mgnify_get_analyses_results(...)

mgnify_get_analyses_phyloseq(...)

mgnify_get_analyses_metadata(...)

mgnify_retrieve_json(...)
```

Arguments

username, password	Webin credentials for mgnifyLogin.
useCache, cacheDir	If useCache = TRUE, cacheDir is set as the cache directory.
...	In doQuery, query parameters of the API. Otherwise ignored.

x	Ignored. Formerly a MgnifyClient object.
accession	A character vector of accessions.
get.taxa	Taxonomy to fetch, e.g., "SSU".
get.func, output	Not supported anymore.
type	The type of accessions in searchAnalysis and searchFile, or the API endpoint in doQuery and getData, e.g., "studies".
as.df	Whether to return a data.frame instead of a list.
max.hits	The maximum number of results, or NULL for all.
accession.type	The type of accessions in getData.
url	The URL of the file.
file	NULL or a path to save the file to.
read.func	NULL or a function applied to the downloaded file.
value	Ignored.

Details

MGnify API v1 is frozen, and MGnifyR now uses MGnify API v2. The functions of the earlier versions of MGnifyR are deprecated, and they will be removed in a future release. Where possible, they use the new functions:

- MgnifyClient: a client is not needed anymore. Log in with mgnifyLogin to access private data, and set options(MGnifyR.cache = "path") to keep the cache between sessions. For compatibility, MgnifyClient does these, and returns NULL. The x argument of the other deprecated functions is ignored.
- getResult: use fetchAnalyses. Only taxonomic profiles are returned, as TreeSummarizedExperiment.
- searchAnalysis and getMetadata: use searchAnalyses.
- doQuery and getData: use searchStudies, searchSamples or searchAnalyses. These query any endpoint of MGnify API v2, but the endpoints and filters differ from API v1.
- searchFile and getFile: list and download the files of studies, samples or analyses.

The accessors of MgnifyClient, such as useCache, and the functions with mgnify_ prefix are defunct.

Value

See the replacing functions. doQuery, getData and searchFile return a data.frame, and getFile returns the path of the file or the result of read.func.

Examples

```
# Instead of getResult(), use fetchAnalyses()
tse <- fetchAnalyses("MGYA00383350")
```

`fetchStudies` *Fetch taxonomic and functional profiles*

Description

Fetch taxonomic and functional profiles

Usage

```
fetchStudies(
  study.ids,
  type = "SSU",
  samples = FALSE,
  biosamples = FALSE,
  studies = FALSE,
  analyses = FALSE,
  assemblies = FALSE,
  qc = FALSE
)

fetchAnalyses(
  analysis.ids,
  type = "SSU",
  samples = FALSE,
  biosamples = FALSE,
  studies = FALSE,
  analyses = FALSE,
  assemblies = FALSE,
  qc = FALSE
)
```

Arguments

`study.ids`	A character vector specifying study accessions, e.g., `"MGYS00005116"`, or a `data.frame` returned by `searchStudies`.
`type`	A character vector specifying the profiles to fetch, e.g., `"SSU"` or `c("SSU", "GO-slim")`. See details for the types. (By default: `type = "SSU"`)
`samples`	`TRUE` or `FALSE` specifying whether to add the sample metadata in MGnify. (By default: `samples = FALSE`)
`biosamples`	`TRUE` or `FALSE` specifying whether to add the attributes of the samples in BioSamples. (By default: `biosamples = FALSE`)
`studies`	`TRUE` or `FALSE` specifying whether to add the details and publications of the studies. (By default: `studies = FALSE`)
`analyses`	`TRUE` or `FALSE` specifying whether to add the details of the analyses, e.g., the quality control of pipeline v6. (By default: `analyses = FALSE`)

assemblies	TRUE or FALSE specifying whether to add the details of the assemblies. (By default: assemblies = FALSE)
qc	TRUE or FALSE specifying whether to add the quality control of pipelines v1-v5, e.g., the number of sequences before and after quality filtering. It is read from separate files, which makes the search slower. The details of the analyses are added too. (By default: qc = FALSE)
analysis.ids	A character vector specifying analysis accessions, e.g., "MGYA00383350", or a data.frame returned by searchAnalyses. Giving the data.frame is faster.

Details

fetchStudies fetches all the analyses of the given studies, and fetchAnalyses fetches specific analyses, e.g., a filtered result of searchAnalyses. Columns are analyses and rows are taxa or, in functional profiles, functions, e.g., GO terms. Taxa are matched across studies by their taxonomy lineage.

type tells which profile is fetched. Only analyses that have the type are fetched.

- Taxonomic profiles: "SSU" and "LSU" (small and large subunit rRNA), "mOTUs" (marker genes of metagenomes and metatranscriptomes, pipelines v5-v6), "UNITE" and "ITSoneDB" (fungal ITS), "PR2" (eukaryotes), "DADA2-SILVA" and "DADA2-PR2" (amplicon sequence variants, pipeline v6), and "contigs" (number of contigs per taxon in assemblies, pipeline v6). Pipelines v1-v3 have a single profile, which is fetched with "SSU".
- Functional profiles of metagenomes, metatranscriptomes and assemblies: "InterPro", "GO" and "GO-slim" (pipelines v1-v6), "Pfam" and "KO" (KEGG orthologs, pipelines v5-v6), and "antiSMASH" (biosynthetic gene clusters of assemblies, pipelines v5-v6). In pipeline v6, only assemblies have functional profiles.

Profiles of different pipeline versions are based on different reference databases, so they might not be directly comparable. To fetch a single pipeline version, subset the result of searchAnalyses by pipeline_version. Data is cached for the R session; to keep it between sessions, set options(MGnifyR.cache = "path/to/cache").

The arguments samples, biosamples, studies, analyses, assemblies and qc choose the metadata in colData, as in searchAnalyses. In fetchAnalyses, they are used only if analysis.ids are accessions; for a data.frame, give them to searchAnalyses instead.

Value

For a single type, a TreeSummarizedExperiment with

- assay: "counts"
- colData: the columns of searchAnalyses, i.e., the metadata of the analyses, their studies and samples
- rowData: taxonomy ranks, or the descriptions of the rows of functional profiles

For multiple types, a MultiAssayExperiment with a TreeSummarizedExperiment for each type that has data. The profiles are linked by sample, e.g., the SSU profile of an amplicon analysis and the GO profile of a metagenomic analysis of the same sample.

NULL is returned if no data could be fetched.

See Also

searchStudies

Examples

```
# Fetch all analyses of a study
tse <- fetchStudies("MGYS00005116")
tse

# Select analyses and fetch them
analyses <- searchAnalyses("MGYS00005116")
analyses <- analyses[analyses[["pipeline_version"]] == "V5", ]
tse <- fetchAnalyses(analyses)


# Fetch a functional profile of a study
tse <- fetchStudies("MGYS00005610", type = "GO-slim")

# Fetch taxonomic and functional profiles into MultiAssayExperiment
mae <- fetchStudies("MGYS00005610", type = c("SSU", "GO-slim"))

# Browse studies and fetch them
studies <- searchStudies(biome = "root:Host-associated:Human", max.hits = 5)
tse <- fetchStudies(studies)
```

---

mgnifyLogin *Log in to MGnify to access private data*

Description

Log in to MGnify to access private data

Usage

```
mgnifyLogin(username, password)
```

Arguments

username	A single character value specifying the Webin username, e.g., "Webin-1122334".
password	A single character value specifying the password.

Details

Most MGnify data is public and can be fetched without logging in. To access private data, log in with your ENA Webin credentials. The login lasts for the R session, and it is used by all the functions that fetch data from MGnify API v2, such as searchStudies and fetchStudies.

Value

TRUE invisibly, if login was successful.

Examples

```
# Wrong credentials give an error
try(mgnifyLogin("Webin-0", "wrong"))

# With your own ENA Webin credentials, log in and fetch private data
# mgnifyLogin("Webin-1122334", "SecretPassword")
# tse <- fetchStudies("MGYS00000001")
```

---

searchStudies *Search MGnify studies, samples and analyses*

Description

Search MGnify studies, samples and analyses

Usage

```
searchStudies(
  biome = NULL,
  pipeline.version = NULL,
  search = NULL,
  order = NULL,
  max.hits = NULL
)

searchSamples(
  study.ids = NULL,
  biome = NULL,
  search = NULL,
  order = NULL,
  max.hits = NULL
)

searchAnalyses(
  study.ids,
  samples = FALSE,
  biosamples = FALSE,
  studies = FALSE,
  analyses = FALSE,
  assemblies = FALSE,
  qc = FALSE
)
```

Arguments

biome	NULL or a single character value specifying a biome lineage, e.g., "root:Environmental:Aquatic". Results also include the descendant biomes. Matching is not case-sensitive. The available biomes are listed at https://www.ebi.ac.uk/metagenomics/browse/biomes. (By default: biome = NULL)
pipeline.version	NULL or a single character value specifying the version of the MGnify pipeline: "V1", "V2", "V3", "V4", "V4.1", "V5", "V6", "V6.1" or "V6.2". The "V" and ".0" can be omitted, e.g., "5.0" or "5". (By default: pipeline.version = NULL)
search	NULL or a single character value used to search titles and accessions. (By default: search = NULL)
order	NULL or a single character value specifying the order of the results. For studies, "accession" or "updated_at", and for samples, "sample_title" or "updated_at". Prefix with "-" for descending order, e.g., "-updated_at". (By default: order = NULL)
max.hits	NULL or a single integer value specifying the maximum number of results. If NULL, all results are returned. (By default: max.hits = NULL)
study.ids	A character vector specifying study accessions, e.g., "MGYS00005116", or a data.frame returned by searchStudies. In searchSamples, it can also be NULL to search all samples. biome, search and order can be used only when searching all samples.
samples	TRUE or FALSE specifying whether to add the sample metadata in MGnify. (By default: samples = FALSE)
biosamples	TRUE or FALSE specifying whether to add the attributes of the samples in BioSamples. (By default: biosamples = FALSE)
studies	TRUE or FALSE specifying whether to add the details and publications of the studies. (By default: studies = FALSE)
analyses	TRUE or FALSE specifying whether to add the details of the analyses, e.g., the quality control of pipeline v6. (By default: analyses = FALSE)
assemblies	TRUE or FALSE specifying whether to add the details of the assemblies. (By default: assemblies = FALSE)
qc	TRUE or FALSE specifying whether to add the quality control of pipelines v1-v5, e.g., the number of sequences before and after quality filtering. It is read from separate files, which makes the search slower. The details of the analyses are added too. (By default: qc = FALSE)

Details

Browse studies with searchStudies and fetch their data with fetchStudies. To select specific analyses, e.g., by sample metadata or pipeline version, filter the result of searchAnalyses and give it to fetchAnalyses.

searchAnalyses returns one row per analysis with the accessions, pipeline version, experiment type, sequencing instrument, and the other fields of the analysis in the list of analyses of its study. The other sources are added with separate arguments, as each needs its own requests. By default,

none of them is added, which is the fastest. The data is cached, so a later search with more sources fetches only the new data:

- `samples`: sample metadata in MGnify (`sample_*`), one request per 100 samples
- `biosamples`: all attributes of the samples in BioSamples (`biosamples_*`), including study-specific variables, e.g., `biosamples_disease_status`, one request per 100 samples
- `studies`: details and publications of the studies (`study_*`), two requests per study
- `analyses`: details of the analyses (`analysis_*`), e.g., the quality control of pipeline v6, one request per analysis
- `assemblies`: details of the assemblies (`assembly_*`), one request per assembly
- `qc`: quality control files of pipelines v1-v5 (`analysis_*`), several files per analysis, and the details of the analyses

These are the same columns as in the `colData` of `fetchAnalyses`. Values are as in the database. Lists, e.g., the publications of a study (`study_publications`), are JSON text that can be read with `jsonlite::fromJSON()`.

In MGnify, a study has samples, a sample has runs, i.e., sequencings of the sample, and an analysis is a run or an assembly processed with an MGnify pipeline. A sample can have multiple runs, e.g., an amplicon and a metagenome, and a run can be analysed with multiple pipeline versions, so a sample can have multiple analyses. An analysis usually has one run, but some have multiple runs, e.g., `"ERR1, ERR2"`, and analyses of assemblies can have no run in `run_accession`.

Results are cached for the R session; to keep them between sessions, set `options(MGnifyR.cache = "path/to/cache")`. To access private data, log in first with `mgnifyLogin`.

Value

A `data.frame` where each row is a study, sample or analysis.

See Also

`fetchStudies`

Examples

```
# Search studies of aquatic environments
studies <- searchStudies(biome = "root:Environmental:Aquatic", max.hits = 10)

# Get samples and analyses of a study
samples <- searchSamples("MGYS00005116")
analyses <- searchAnalyses("MGYS00005116")
analyses <- analyses[analyses[["pipeline_version"]] == "V6", ]
```

Index

authTok *(deprecated)*, 2
authTok<- *(deprecated)*, 2

cacheDir *(deprecated)*, 2
cacheDir<- *(deprecated)*, 2
clearCache *(deprecated)*, 2
clearCache<- *(deprecated)*, 2

databaseUrl *(deprecated)*, 2
databaseUrl<- *(deprecated)*, 2
deprecated, 2
doQuery *(deprecated)*, 2

fetchAnalyses, *5*, *10, 11*
fetchAnalyses *(fetchStudies)*, 6
fetchStudies, *2*, 6, *8*, *10, 11*

getData *(deprecated)*, 2
getFile *(deprecated)*, 2
getMetadata *(deprecated)*, 2
getResult *(deprecated)*, 2

mgnify_analyses_from_samples *(deprecated)*, 2
mgnify_analyses_from_studies *(deprecated)*, 2
mgnify_client *(deprecated)*, 2
mgnify_download *(deprecated)*, 2
mgnify_get_analyses_metadata *(deprecated)*, 2
mgnify_get_analyses_phyloseq *(deprecated)*, 2
mgnify_get_analyses_results *(deprecated)*, 2
mgnify_get_download_urls *(deprecated)*, 2
mgnify_query *(deprecated)*, 2
mgnify_retrieve_json *(deprecated)*, 2
MgnifyClient *(deprecated)*, 2
mgnifyLogin, *4, 5*, 8, *11*
MGnifyR *(MGnifyR-package)*, 2
MGnifyR-package, 2

searchAnalyses, *5*, *7*
searchAnalyses *(searchStudies)*, 9
searchAnalysis *(deprecated)*, 2
searchFile *(deprecated)*, 2
searchSamples, *5*
searchSamples *(searchStudies)*, 9
searchStudies, *2*, *5, 6*, *8*, 9
showWarnings *(deprecated)*, 2
showWarnings<- *(deprecated)*, 2

TreeSummarizedExperiment, *2*

useCache *(deprecated)*, 2
useCache<- *(deprecated)*, 2

verbose *(deprecated)*, 2
verbose<- *(deprecated)*, 2