

Package ‘modisfast’

October 5, 2026

Title Fast and Efficient Access to MODIS Earth Observation Data

Version 2.0.1

Description Programmatic interface to several NASA Earth Observation 'OPeNDAP' servers (Open-source Project for a Network Data Access Protocol) (<<https://www.opendap.org/>>). Allows for easy downloads of MODIS subsets, as well as other Earth Observation data-cubes, in a time-saving and efficient way : by sampling it at the very downloading phase (spatially, temporally and dimensionally).

License GPL (>= 3)

URL <https://github.com/ptaconet/modisfast>

BugReports <https://github.com/ptaconet/modisfast/issues>

Depends R (>= 3.5)

Imports curl, dplyr, httr, jsonlite, lubridate, magrittr, parallel,
purrr, rvest, sf, stringr, terra, xml2, cli

Suggests ggplot2, knitr, mapview, rmarkdown, spelling, testthat

Encoding UTF-8

LazyData true

Language en-US

Config/testthat/start-first mf_list_collections, mf_list_variables,
mf_get_url, mf_download_data, mf_import_data

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Paul Taconet [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-7429-7204>>),
Nicolas Moiroux [fnd] (ORCID: <<https://orcid.org/0000-0001-6755-6167>>),
French National Research Institute for Sustainable Development, IRD
[fnd]

Maintainer Paul Taconet <paul.taconet@gmail.com>

Repository CRAN

Date/Publication 2026-10-04 23:10:02 UTC

Contents

entomological_data	2
mf_download_data	3
mf_get_url	5
mf_import_data	6
mf_list_collections	8
mf_list_collections_cloud	8
mf_list_variables	9
mf_modisfast	11
Index	13

entomological_data	<i>Example dataset containing abundances of mosquitoes vectors of malaria. Used in article 'use_case'.</i>
--------------------	--

Description

Example dataset containing abundances of mosquitoes vectors of malaria. Used in article 'use_case'.

Usage

entomological_data

Format

'entomological_data' A data frame with 232 rows and 6 columns:

mission number of the entomological survey

date date of the survey

village 3-digit code for the village of the survey

X, Y longitude and latitude of the center of the village

n number of mosquitoes collected

Source

<<https://doi.org/10.15468/v8fvyn>>

mf_download_data	<i>Download several datasets given their URLs and destination path</i>
------------------	--

Description

This function enables to download datasets. In a data import workflow, this function is typically used after a call to the [mf_get_url](#) function. The output value of [mf_get_url](#) can be used as input of parameter `df_to_dl` of [mf_download_data](#).

The download can be parallelized.

Usage

```
mf_download_data(
  df_to_dl,
  path = tempfile("modisfast_"),
  parallel = FALSE,
  num_workers = parallel::detectCores() - 1,
  verbose = "inform",
  min_filesize = 5000
)
```

Arguments

<code>df_to_dl</code>	data.frame. Urls and destination files of dataset to download. Typically output of mf_get_url . See Details for the structure
<code>path</code>	string. Target folder for the data to download. Default : temporary folder.
<code>parallel</code>	boolean. Parallelize the download ? Default to FALSE
<code>num_workers</code>	integer. Number of workers in case of parallel download. Default to number of workers available in the machine minus one.
<code>verbose</code>	Character string: "quiet", "inform" (default), or "debug". Controls progress messages; "debug" also shows request details.
<code>min_filesize</code>	integer. Minimum file size expected (in bites) for one file downloaded. If files downloaded are less than this value, the files will be downloaded again. Default 5000.

Details

Parameter `df_to_dl` must be a data.frame with the following minimal structure :

id_roi An id for the ROI (character string)

collection Collection (character string)

name

url URL of the file to download (character string)

Value

a data.frame with the same structure of the input data.frame df_to_dl + columns providing details of the data downloaded. The additional columns are :

fileDI Boolean (dataset downloaded or failure)

dlStatus Download status : 1 = download ok ; 2 = download error ; 3 = dataset was already existing in destination file

fileSize File size on disk (in bites)

Examples

```
## Not run:

### Configure an Earthdata bearer token for LP DAAC Cloud
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")

### Set-up parameters of interest
coll <- "VJ121A2.002"

bands <- c("LST_Day_1KM", "LST_Night_1KM")

time_range <- as.Date(c("2026-01-01", "2026-01-30"))

roi <- sf::st_as_sf(
  data.frame(
    id = "roi_test",
    geom = "POLYGON ((-5.82 9.54, -5.42 9.55, -5.41 8.84, -5.81 8.84, -5.82 9.54))"
  ),
  wkt = "geom", crs = 4326
)

### Get the URLs of the data
(urls_vj121a2 <- mf_get_url(
  collection = coll,
  variables = bands,
  roi = roi,
  time_range = time_range
))

### Download the data
res_dl <- mf_download_data(urls_vj121a2)

### Import the data as terra::SpatRast
modis_ts <- mf_import_data(dirname(res_dl$destfile[1]), collection = coll)

### Plot the data
terra::plot(modis_ts)

## End(Not run)
```

mf_get_url

*Find download URLs for MODIS, VIIRS and GPM data***Description**

Find the files covering your area and dates, and create URLs to download only the selected bands and area. Use [mf_download_data()] to download them.

Usage

```
mf_get_url(
  collection,
  variables = NULL,
  roi,
  time_range,
  collection_id = NULL,
  verbose = "inform"
)
```

Arguments

collection	Collection identifier, e.g. "MOD11A1.061", "VNP43MA4.002" or "GPM_3IMERGDF.07".
variables	One or more band names to retrieve, as returned by [mf_list_variables()].
roi	An 'sf' polygon with an 'id' column.
time_range	One date or two bounding dates.
collection_id	Optional Earthdata Cloud collection ID. Usually leave this empty; the package finds it in its collection catalogue. Not used for GPM.
verbose	Character string: "quiet", "inform" (default), or "debug". Controls progress messages; "debug" also shows request details.

Details

Set 'EARTHDATA_TOKEN' to your Earthdata token before calling this function.

Value

A data frame with the download URLs for [mf_download_data()].

Examples

```
## Not run:
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")
roi <- sf::st_as_sf(data.frame(id = "test", geom =
  "POLYGON ((3.8 43.5, 4 43.5, 4 43.7, 3.8 43.7, 3.8 43.5))"),
  wkt = "geom", crs = 4326)
time_range <- as.Date(c("2026-01-01", "2026-01-30"))
urls_vj121a2 <- mf_get_url("VJ121A2.002",
```

```
c("LST_Day_1KM", "LST_Night_1KM"), roi, time_range)
mf_download_data(urls_vj121a2)

## End(Not run)
```

mf_import_data	<i>Import datasets downloaded using modisfast as a terra::SpatRaster object</i>
----------------	---

Description

Import datasets downloaded using modisfast as a terra::SpatRaster object

Usage

```
mf_import_data(
  path,
  collection,
  output_class = "SpatRaster",
  proj_epsg = NULL,
  roi_mask = NULL,
  vrt = FALSE,
  verbose = "inform",
  ...
)
```

Arguments

path	Download root folder supplied to [mf_download_data()], or a folder containing the NetCDF files for one ROI and collection.
collection	Collection identifier, e.g. "MOD11A1.061", "VNP43MA4.002" or "GPM_3IMERGDF.07".
output_class	character string. Output object class. Currently only "SpatRaster" implemented.
proj_epsg	numeric. EPSG of the desired projection for the output raster (default : source projection of the data).
roi_mask	SpatRaster or SpatVector or sf. Area beyond which data will be masked. Typically, the input ROI of mf_get_url (default : NULL (no mask))
vrt	boolean. Import virtual raster instead of SpatRaster. Useful for very large files. (default : FALSE)
verbose	Character string: "quiet", "inform" (default), or "debug". Controls progress messages.
...	not used

Value

a terra::SpatRast object

Note

Although the data downloaded through `modisfast` could be imported with any netcdf-compliant R package (`terra`, `stars`, `ncdf4`, etc.), care must be taken. In fact, depending on the collection, some “issues” were raised. These issues are independent from `modisfast` : they result most of time of a lack of full implementation of the OPeNDAP framework by the data providers. Namely, these issues are :

- for MODIS and VIIRS collections : CRS has to be provided
- for GPM collections : EPSG:4326 is assigned to the grid

The function `mf_import_data` includes the processing that needs to be done at the data import phase in order to safely use the data as `terra` objects.

Also note that reprojecting over large ROIs using the argument `proj_epsg` might take long. In this case, setting the argument `vrt` to `TRUE` might be a solution.

Examples

```
## Not run:

### Configure an Earthdata bearer token for LP DAAC Cloud
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")

### Set-up parameters of interest
coll <- "VJ121A2.002"

bands <- c("LST_Day_1KM", "LST_Night_1KM")

time_range <- as.Date(c("2026-01-01", "2026-01-30"))

roi <- sf::st_as_sf(
  data.frame(
    id = "roi_test",
    geom = "POLYGON ((-5.82 9.54, -5.42 9.55, -5.41 8.84, -5.81 8.84, -5.82 9.54))"
  ),
  wkt = "geom", crs = 4326
)

### Get the URLs of the data
(urls_vj121a2 <- mf_get_url(
  collection = coll,
  variables = bands,
  roi = roi,
  time_range = time_range
))

### Download the data
res_dl <- mf_download_data(urls_vj121a2)

### Import the data as terra::SpatRast
modis_ts <- mf_import_data(dirname(res_dl$destfile[1]), collection = coll)
```

```
### Plot the data
terra::plot(modis_ts)

## End(Not run)
```

mf_list_collections	<i>Get the collections available for download with the modisfast package</i>
---------------------	--

Description

Get the collections available for download using the package and a set of related information

Usage

```
mf_list_collections()
```

Value

A data.frame with the collections available, and a set of related information for each one. Main columns are :

collection Collection short name

source Data provider

long_name Collection long name

doi DOI of the collection

start_date First available date for the collection

url_opendapserver URL of the OPeNDAP server of the data

Examples

```
(head(mf_list_collections()))
```

mf_list_collections_cloud	<i>Check LP DAAC MODIS and VIIRS collections on Earthdata Cloud OPeNDAP</i>
---------------------------	---

Description

CMR is paginated to discover every LPCLOUD collection supported by modisfast. For each collection, the function tests up to 'n_granules' granules (newest first), including a single pixel data request. A successful sample does not imply that every granule or date is available.

Usage

```
mf_list_collections_cloud(
  collections = NULL,
  time_range = NULL,
  n_granules = 2L,
  verbose = "inform"
)
```

Arguments

<code>collections</code>	Optional character vector of collection names, for example <code>c("MOD11A1.061", "VNP43MA4.002")</code> . By default check LP DAAC MODIS collections from the modisfast catalogue and tiled VIIRS <code>‘.002’</code> collections published by LPCLOUD in CMR.
<code>time_range</code>	Optional one or two Dates restricting sampled granules.
<code>n_granules</code>	Number of granules to check per collection (default 2).
<code>verbose</code>	Character string: <code>"quiet"</code> , <code>"inform"</code> (default), or <code>"debug"</code> . Controls progress messages.

Value

A data.frame with `collection`, `collection_id`, `status`, `tested`, `succeeded`, `checked_at`, and `detail`. Status is `‘operational’` when all sampled granules served one pixel, `‘partial’` when only some did, `‘unavailable’` when none did, and `‘no_granules’` when CMR found none for the chosen dates. `‘unverified’` means no suitable two-dimensional variable was found. These statuses apply only to the granules sampled at `‘checked_at’`.

Examples

```
## Not run:
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")
status <- mf_list_collections_cloud("VJ121A2.002")
subset(status, status == "operational")
mf_list_collections_cloud("VJ121A2.002",
  time_range = as.Date(c("2026-01-01", "2026-01-30")))

## End(Not run)
```

<code>mf_list_variables</code>	<i>Get information for the variables (bands) available for a given collection</i>
--------------------------------	---

Description

Get the variables available for a given collection, along with a set of related information for each.

Usage

```
mf_list_variables(
  collection,
  verbose = "inform",
  backend = c("auto", "cloud", "legacy"),
  time_range = NULL,
  collection_id = NULL,
  n_granules = 10L
)
```

Arguments

collection	Collection identifier, e.g. "MOD11A1.061", "VNP43MA4.002" or "GPM_3IMERGDF.07".
verbose	Character string: "quiet", "inform" (default), or "debug". Controls progress messages; "debug" also shows request details.
backend	"auto" (default) chooses the service for the collection. "cloud" selects Earthdata Cloud; "legacy" selects GPM's GES DISC service.
time_range	Optional date or pair of dates used to find a Cloud granule. Only used for Cloud collections.
collection_id	Optional Earthdata Cloud collection ID. Usually leave this empty; the package looks it up. Only used for Cloud collections.
n_granules	Maximum number of recent Cloud files to try (default 10).

Value

A data.frame with the variables available for the collection, and a set of related information for each variable. The variables marked as "extractable" in the column "extractable_with_modisfast" can be provided as input parameter variables of `mf_get_url`. Cloud metadata comes from one accessible granule's DDS; it lists its two-dimensional fields but does not supply long names or units.

Examples

```
## Not run:
# Configure an Earthdata bearer token for Cloud OPeNDAP.
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")

time_range <- as.Date(c("2026-01-01", "2026-01-30"))
bands <- mf_list_variables("VJ121A2.002", time_range = time_range)
subset(bands, name %in% c("LST_Day_1KM", "LST_Night_1KM"))

## End(Not run)
```

mf_modisfast	<i>Download (and possibly import) MODIS, VIIRS and GPM Earth Observation data</i>
--------------	---

Description

Download and possibly import MODIS, VIIRS and GPM Earth Observation data quickly and efficiently. This function is a wrapper for [mf_get_url](#), [mf_download_data](#) and [mf_import_data](#). Whenever possible, users should prefer executing the functions [mf_get_url](#), [mf_download_data](#) and [mf_import_data](#) sequentially rather than using this high-level function

Usage

```
mf_modisfast(
  collection,
  variables,
  roi,
  time_range,
  path = tempfile("modisfast_"),
  parallel = FALSE,
  verbose = "inform",
  import = TRUE,
  ...
)
```

Arguments

collection	Collection identifier, e.g. "MOD11A1.061", "VNP43MA4.002" or "GPM_3IMERGDF.07".
variables	One or more band names to retrieve, as returned by [mf_list_variables()] .
roi	An 'sf' polygon with an 'id' column.
time_range	One date or two bounding dates.
path	string. Target folder for the data to download. Default : temporary folder.
parallel	boolean. Parallelize the download ? Default to FALSE
verbose	Character string: "quiet", "inform" (default), or "debug". Controls progress messages; "debug" also shows request details.
import	boolean. Import the data as a SpatRast object ? default TRUE. FALSE will download the data but not import them in R.
...	Further arguments to be passed to mf_import_data

Value

if the parameter import is set to TRUE, a terra::SpatRast object ; else a data.frame providing details of the data downloaded (see output of [mf_download_data](#)).

See Also

[mf_get_url](#), [mf_download_data](#), [mf_import_data](#)

Examples

```
## Not run:

### Set-up parameters of interest
coll <- "VJ121A2.002"

bands <- c("LST_Day_1KM", "LST_Night_1KM")

time_range <- as.Date(c("2026-01-01", "2026-01-30"))

roi <- sf::st_as_sf(
  data.frame(
    id = "roi_test",
    geom = "POLYGON ((-5.82 9.54, -5.42 9.55, -5.41 8.84, -5.81 8.84, -5.82 9.54))"
  ),
  wkt = "geom", crs = 4326
)

### Download and import the data
Sys.setenv(EARTHDATA_TOKEN = "your Earthdata bearer token")
modis_ts <- mf_modisfast(
  collection = coll,
  variables = bands,
  roi = roi,
  time_range = time_range
)

### Plot the data
terra::plot(modis_ts)

## End(Not run)
```

Index

* **datasets**

entomological_data, [2](#)

entomological_data, [2](#)

mf_download_data, [3](#), [3](#), [11](#), [12](#)

mf_get_url, [3](#), [5](#), [6](#), [10–12](#)

mf_import_data, [6](#), [7](#), [11](#), [12](#)

mf_list_collections, [8](#)

mf_list_collections_cloud, [8](#)

mf_list_variables, [9](#)

mf_modisfast, [11](#)