Package: datalift (via r-universe)

November 13, 2026

Title Territorial data acquisition

Version 0.1.0

Description **DATALIFT** (Territorial Data Acquisition) is an R package designed to facilitate the acquisition, download and preparation of spatial and environmental datasets from multiple online data providers. It provides a unified interface to access and process data related to topography, land cover, soils, meteorology, LiDAR and satellite imagery. DATALIFT integrates several open-source tools and packages, including 'happign', 'lidarHD' and 'RADIS', to support reproducible territorial data acquisition workflows.

License GPL (>= 3)

URL https://fabricevinatierpro-hub.r-universe.dev/datalift

BugReports https://fabricevinatierpro-hub.r-universe.dev/datalift/issues

Imports config, dplyr, golem, happign, httr, leafem, leaflet, leaflet.extras, lidarHD, lubridate, openeo, osmdata, pbapply, RADIS, readr, sf, shiny, shinyjs, stats, terra, tibble, tools, utils

Suggests testthat (>= 3.0.0)

Remotes lidarHD=git::https://forge.inrae.fr/lidar/lidarHD.git

Additional_repositories https://fabricevinatierpro-hub.r-universe.dev

Config/roxygen2/version 8.1.1

Encoding UTF-8

LazyData true

Config/pak/sysreqs libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev libglu1-mesa-dev make libharfbuzz-dev texlive libarchive-dev libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libgl1-mesa-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev zlib1g-dev

Repository https://fabricevinatierpro-hub.r-universe.dev

Date/Publication 2026-11-13 15:44:08 UTC

RemoteUrl https://forge.inrae.fr/fabrice.vinatier/datalift.git

RemoteRef HEAD

RemoteSha 6795a1a254a473254828a3827e6627042a7da12a

Contents

bytes_to_human . . . 3
calc_duration_size . . . 3
export_wms_from_sf_fixed_res . . . 4
fct.read.meteofrance . . . 5
fit_model . . . 7
folder_size . . . 7
get_duration_size_models . . . 8
get_sources . . . 9
load_AWC . . . 9
load_BDTOPO . . . 10
load_CLC . . . 10
load_DEPTH . . . 11
load_ISBA . . . 11
load_LIDAR . . . 12
load_METEO . . . 12
load_MNE . . . 13
load_OCS . . . 14
load_ORTHO . . . 14
load_OSM . . . 15
load_PCRS . . . 15
load_RPG . . . 16
load_RPG_supplement . . . 16
load_SENTINEL . . . 17
load_SOL . . . 18
load_stations_metadatas . . . 19
load_TEXTURE . . . 20
load_UAV . . . 20
run_app . . . 21
seconds_to_human . . . 22
select_METEO_stations . . . 22

Index **24**

bytes_to_human *Convert bytes to human-readable units*

Description

Express a file size in the most appropriate binary unit, from bytes to petabytes.

Usage

```
bytes_to_human(bytes, digits = 1)
```

Arguments

bytes	Numeric scalar. Non-negative file size in bytes.
digits	Integer. Number of decimal places used for rounding.

Value

Character string containing the rounded value and its unit (‘B‘, ‘Ko‘, ‘Mo‘, ‘Go‘, ‘To‘ or ‘Po‘).

See Also

[folder_size()]

Examples

```
bytes_to_human(1536)
bytes_to_human(1048576, digits = 2)
```

calc_duration_size *Benchmark download duration and output size*

Description

Run the loader associated with a dataset name, measure its elapsed execution time and calculate the size of the corresponding output folder.

Usage

```
calc_duration_size(dataset = "MNE", vWIN, path_export, delete_files = TRUE)
```

Arguments

dataset	Character string identifying the dataset and corresponding loader function. For example, ‘"MNE"‘ calls ‘load_MNE()‘.
vWIN	Spatial object passed to the loader as the study area.
path_export	Character string. Root output directory passed to the loader.
delete_files	Logical. If ‘TRUE‘, remove ‘path_export‘ recursively after the benchmark. Defaults to ‘TRUE‘.

Details

The loader is resolved by constructing the function name ‘load_<dataset>‘. The output size is measured in the subdirectory named after ‘dataset‘. Use a dedicated temporary output directory when ‘delete_files = TRUE‘.

Value

Numeric vector of length two: elapsed time in seconds and output size in bytes.

Examples

```
## Not run:
area <- sf::st_read("study_area.gpkg")
calc_duration_size("MNE", area, tempfile())

## End(Not run)
```

---

export_wms_from_sf_fixed_res

Export a WMS raster at fixed resolution using spatial tiles

Description

Divide a spatial bounding box into tiles and download each tile from a WMS service at a fixed map resolution. Each tile is exported as JPEG2000.

Usage

```
export_wms_from_sf_fixed_res(
  sf_obj,
  layer,
  crs_out = 2154,
  resolution = 0.05,
  max_pixels = 5000,
  outdir = "tiles"
)
```

Arguments

sf_obj	An ‘sf‘ or ‘sfc‘ spatial object defining the export extent.
layer	Character string. WMS layer identifier understood by the configured service.
crs_out	Coordinate reference system used to interpret the bounding box. Defaults to EPSG:2154.
resolution	Numeric. Requested raster resolution in units of ‘crs_out‘. Defaults to 0.05.
max_pixels	Integer. Maximum number of pixels along each tile dimension. Defaults to 5000.
outdir	Character string. Output directory for the JPEG2000 tiles.

Details

The maximum nominal tile width and height are each ‘max_pixels * resolution‘ map units. The output uses the GDAL ‘JP2OpenJPEG‘ driver with lossless encoding options (‘QUALITY=100‘ and ‘REVERSIBLE=YES‘). Availability of that GDAL driver is required.

Value

The function writes files named ‘tile_<n>.jp2‘ to ‘outdir‘ and returns no consolidated raster object.

See Also

[load_PCRS()]

Examples

```
## Not run:
area <- sf::st_read("study_area.gpkg")
export_wms_from_sf_fixed_res(
  area,
  layer = "PCRS.LAMB93",
  resolution = 0.05,
  outdir = "OUT/PCRS"
)

## End(Not run)
```

---

fct.read.meteofrance *Retrieve Météo-France observations for one station*

Description

Authenticate with the Météo-France API, submit observation-data requests for successive calendar years and combine the returned tables.

Usage

```
fct.read.meteofrance(
  auth_base64 = Sys.getenv("METEOFRANCE_AUTH_BASE64"),
  interval = "daily",
  station_id = "34237001",
  from_date = "2000-01-01",
  to_date = "2010-12-31"
)
```

Arguments

auth_base64	Character string. Base64-encoded API credentials. By default, read from the 'METEOFRANCE_AUTH_BASE64' environment variable.
interval	Character string. Observation interval: '"minutely"', '"hourly"' or '"daily"'.
station_id	Character string. Météo-France station identifier.
from_date	Character string. First observation date in 'YYYY-MM-DD' format.
to_date	Character string. Last observation date in 'YYYY-MM-DD' format.

Details

The function uses the OAuth2 client-credentials flow. It submits one API command per calendar year, waits five seconds before attempting to retrieve each result, and removes columns containing only missing values. API credentials must not be hard-coded in scripts or package source files.

Value

A data frame containing the observations returned by the API for the requested station and period.

See Also

[load_METEO()], [select_METEO_stations()]

Examples

```
## Not run:
observations <- fct.read.meteofrance(
  interval = "daily", station_id = "34237001",
  from_date = "2020-01-01", to_date = "2020-12-31"
)

## End(Not run)
```

fit_model — *Fit a simple duration or size model*

Description

Fit a linear, exponential or power relationship between a predictor and response, and return the fitted coefficients.

Usage

```
fit_model(x, y, type = "exp")
```

Arguments

x	Numeric vector of predictor values.
y	Numeric vector of response values. For ‘type = "power"‘ or ‘type = "exp"‘, values must be positive because the response is log-transformed.
type	Character string selecting the model: ‘"lin"‘ for a linear model, ‘"exp"‘ for an exponential model, or ‘"power"‘ for a power law.

Value

Numeric vector of length two containing coefficients ‘a‘ and ‘b‘. The corresponding model forms are ‘y = a + b*x‘ (‘lin‘), ‘y = a*exp(b*x)‘ (‘exp‘) and ‘y = a*x^b‘ (‘power‘).

Examples

```
fit_model(c(1, 2, 3, 4), c(2, 4, 6, 8), type = "lin")
```

folder_size — *Calculate the total size of a directory*

Description

Recursively list the files contained in a directory and sum their sizes.

Usage

```
folder_size(path)
```

Arguments

path	Character string. Path to an existing directory.

Details

File sizes that cannot be read are ignored through ‘na.rm = TRUE‘.

Value

Numeric scalar giving the total size of the listed files in bytes.

See Also

[bytes_to_human()]

Examples

```
## Not run:
folder_size("OUT")
bytes_to_human(folder_size("OUT"))

## End(Not run)
```

`get_duration_size_models` *Download duration and size estimation models*

Description

Download duration and size estimation models

Usage

```
get_duration_size_models()
```

Value

A named list containing Duration and Size model functions.

get_sources *Available data sources*

Description

Available data sources

Usage

```
get_sources()
```

Value

A named list describing the supported data sources.

load_AWC *Get Soil Available Water Capacity (AWC)*

Description

Retrieves soil available water capacity (AWC) spatial data from BDGSF and Info&Sols sources.

Usage

```
load_AWC(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory for exported data.

Value

The function writes AWC files to disk.

load_BDTOPO *Download selected BD TOPO vector layers*

Description

Retrieve selected topographic, hydrographic, vegetation, building and administrative layers from the IGN BD TOPO WFS service.

Usage

```
load_BDTOPO(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes available vector layers to GeoPackage files.

load_CLC *Download CORINE Land Cover vector layers*

Description

Retrieve French CORINE Land Cover layers.

Usage

```
load_CLC(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes available CLC vector layers.

load_DEPTH *Get Soil Depth for a Study Area*

Description

Retrieves soil depth data for a given study area. The function supports multiple data sources, with "Base de Données Géographique des Sols de France" (BDGSF) as the default source.

Usage

```
load_DEPTH(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory for exported data.

Value

The function writes DEPTH files to disk.

load_ISBA *Retrieve SIM2 daily weather data from API*

Description

Retrieves daily SIM2 data via the G-EAU API.

Usage

```
load_ISBA(vWIN, path_export, from_date = "2023-01-01", to_date = "2023-01-31")
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory for exported data.
from_date	Character string. Start date (YYYY-MM-DD).
to_date	Character string. End date (YYYY-MM-DD).

Value

The function writes an ISBA csv file to disk.

load_LIDAR *Download classified LiDAR HD point-cloud tiles*

Description

Retrieve the LiDAR HD tile index intersecting a study area and download the corresponding classified point-cloud files.

Usage

```
load_LIDAR(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root output directory.

Value

The function writes tile metadata and downloaded point-cloud files.

load_METEO *Download Météo-France station observations*

Description

Select meteorological stations relevant to a study area and observation period, then download observations for each selected station.

Usage

```
load_METEO(
  vWIN,
  path_export,
  interval = "daily",
  from_date = "2020-01-01",
  to_date = as.character(Sys.Date())
)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the study area.
path_export	Character string. Root directory in which the ‘METEO‘ folder, station metadata and observation CSV files are written.
interval	Character string. Observation interval: ‘"minutely"‘, ‘"hourly"‘ or ‘"daily"‘.
from_date	Character string. First observation date in ‘YYYY-MM-DD‘ format.
to_date	Character string. Last observation date in ‘YYYY-MM-DD‘ format. Defaults to the current date.

Details

API credentials are read from the ‘METEOFRANCE_AUTH_BASE64‘ environment variable. Station selection is performed by [select_METEO_stations()], and observations are retrieved with [fct.read.meteofrance()].

Value

The function writes station metadata (GeoPackage and CSV) and one observation CSV file per selected station. It is primarily used for its side effects.

See Also

[select_METEO_stations()], [fct.read.meteofrance()]

Examples

```
## Not run:
area <- sf::st_read("study_area.gpkg")
load_METEO(area, "OUT", interval = "daily",
           from_date = "2020-01-01")

## End(Not run)
```

---

load_MNE *Download elevation models from IGN services*

Description

Retrieve elevation rasters over a study area from IGN WMS services. The function downloads the RGE ALTI high-resolution elevation model and attempts to retrieve LiDAR HD elevation products (MNH, MNS and MNT).

Usage

```
load_MNE(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory in which the ‘MNE‘ output folder is created.

Value

The function writes GeoTIFF elevation rasters to disk.

---

load_OCS *Download large-scale land-cover and land-use rasters*

Description

Retrieve OCS GE WMTS raster layers.

Usage

```
load_OCS(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes available JPEG2000 rasters.

---

load_ORTHO *Download current and historical IGN orthophotos*

Description

Retrieve current RGB and infrared-colour orthophotos.

Usage

```
load_ORTHO(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes JPEG2000 orthophoto rasters to disk.

load_OSM *Download OpenStreetMap tracks and paths*

Description

Query OpenStreetMap for line features tagged as ‘highway=track‘ or ‘highway=path‘ within the bounding box of a study area.

Usage

```
load_OSM(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root output directory.

Value

The function writes an EPSG:2154 GeoPackage.

load_PCRS *Download large-scale PCRS raster tiles*

Description

Retrieve the Plan Corps de Rue Simplifié (PCRS) WMS layer.

Usage

```
load_PCRS(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes JPEG2000 tiles to disk.

load_RPG *Download agricultural parcel and land-use data*

Description

Retrieve current and historical French agricultural parcel data.

Usage

```
load_RPG(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes agricultural and cadastral vector datasets.

load_RPG_supplement *Download supplemented RPG from ODR services*

Description

The ODR produces a vector geographic layer called "Supplemented RPG" that identifies and fills in missing agricultural parcels in the official RPG using data from IGN and OSO/Cesbio.

Usage

```
load_RPG_supplement(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory for exported data.

Value

The function writes RPG files to disk.

load_SENTINEL *Download Sentinel-1 or Sentinel-2 imagery through openEO*

Description

Build and execute an openEO batch job to retrieve Copernicus Sentinel satellite data over a spatial and temporal extent.

Usage

```
load_SENTINEL(
  vWIN,
  path_export,
  from_date = "2016-01-01",
  to_date = as.character(Sys.Date()),
  ref = "SENTINEL1_GRD",
  con = NULL,
  poll_interval = 60,
  timeout = 21600,
  skip_existing = TRUE
)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest. Its bounding box is used as the openEO spatial extent. If available, ‘vWIN$id[1]‘ is used as the job title.
path_export	Character string. Root output directory. Results are downloaded under ‘SENTINEL/<ref>‘.
from_date	Character string. First acquisition date in ‘YYYY-MM-DD‘ format.
to_date	Character string. Last acquisition date in ‘YYYY-MM-DD‘ format. Defaults to the current date.
ref	Character string. Collection identifier: ‘"SENTINEL1_GRD"‘ or ‘"SENTINEL2_L2A"‘.
con	An authenticated openEO connection returned by ‘openeo::login()‘. Required.
poll_interval	Numeric. Time in seconds between job status checks. Defaults to 60 seconds.
timeout	Numeric. Maximum waiting time in seconds. Defaults to 21600 seconds (6 hours).
skip_existing	Logical. If ‘TRUE‘, avoids downloading dates already covered by existing results. Defaults to ‘TRUE‘.

Details

Sentinel-2 requests the B01–B09, B11, B12 and SCL bands and calculates NDVI. Sentinel-1 requests VH and VV and applies the ‘sar_backscatter‘ process with ‘sigma0-ellipsoid‘ coefficient and noise removal.

An authenticated openEO session is required. The function uses the explicitly supplied connection throughout the entire workflow.

Jobs are monitored using their unique identifiers. The function stops if the job fails, is cancelled, or exceeds the specified timeout.

Value

A character vector containing the paths of downloaded files, or an informative message if no download is necessary.

Examples

```
## Not run:
con <- openeo::connect(
  host = "https://openeo.dataspace.copernicus.eu"
)
con <- openeo::login(con)

area <- sf::st_read("study_area.gpkg")

load_SENTINEL(
  vWIN = area,
  path_export = "OUT",
  from_date = "2024-01-01",
  to_date = "2024-12-31",
  ref = "SENTINEL2_L2A",
  con = con
)

## End(Not run)
```

---

load_SOL *Download soil and land-cover datasets*

Description

Retrieve the INRA soil map and COSIA land-cover rasters.

Usage

```
load_SOL(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object.
path_export	Character string. Root output directory.

Value

The function writes soil and land-cover datasets.

`load_stations_metadatas` *Download and cache Météo-France station metadata*

Description

Download the Météo-France station metadata file when it is missing or older than a specified number of days.

Usage

```
load_stations_metadatas(
  delay = 7,
 url = "https://www.data.gouv.fr/api/1/datasets/r/5ec141e4-3650-4365-82b1-db459efe690c",
  dest = file.path(tools::R_user_dir("datalift", which = "data"),
    "Carte_stations_meteo.csv")
)
```

Arguments

`delay`	Numeric. Maximum age of the cached file, in days, before a fresh copy is downloaded. Defaults to 7.
`url`	Character string. URL of the station metadata resource.
`dest`	Character string. Local destination path. By default, the file is stored in the user data directory for ‘datalift‘.

Details

The destination directory is created when necessary. The default cache location is obtained with ‘tools::R_user_dir("datalift", which = "data")‘.

Value

Character string giving the local path to the station metadata CSV file.

Examples

```
## Not run:
station_file <- load_stations_metadatas()
utils::read.csv(station_file)

## End(Not run)
```

load_TEXTURE *Retrieve soil texture information for spatial features*

Description

These functions retrieve soil texture information for a given spatial feature (sf) object from a specified data source.

Usage

```
load_TEXTURE(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root directory for exported data.

Value

The function writes TEXTURE files to disk.

load_UAV *Download drone flight restriction map*

Description

Retrieve the IGN WMTS layer describing restrictions for open-category unmanned aircraft systems and model aircraft.

Usage

```
load_UAV(vWIN, path_export)
```

Arguments

vWIN	An ‘sf‘ or ‘sfc‘ spatial object defining the area of interest.
path_export	Character string. Root output directory.

Value

The function writes a JPEG2000 raster to disk.

run_app *Run the Shiny Application*

Description

Run the Shiny Application

Usage

```
run_app(
  onStart = NULL,
  options = list(),
  enableBookmarking = NULL,
  uiPattern = "/",
  ...
)
```

Arguments

onStart	A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose.
options	Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app.
enableBookmarking	Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking(). See enableBookmarking() for more information on bookmarking your app.
uiPattern	A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful.
...	arguments to pass to golem_opts. See '?golem::get_golem_options' for more details.

seconds_to_human *Convert seconds to human-readable durations*

Description

Express a duration in the largest applicable unit, from seconds to years.

Usage

```
seconds_to_human(seconds, digits = 1)
```

Arguments

seconds	Numeric scalar. Non-negative duration in seconds.
digits	Integer. Number of decimal places used for rounding.

Value

Character string containing the rounded duration and its unit (‘sec‘, ‘min‘, ‘hours‘, ‘days‘, ‘week‘ or ‘year‘). A year is defined as 365.25 days.

Examples

```
seconds_to_human(90)
seconds_to_human(86400 * 10, digits = 2)
```

select_METEO_stations *Select Météo-France stations for a study area*

Description

Filter the Météo-France station catalogue by department, station type, observation period and temporal resolution. Stations located within the study area are selected; if none are inside, the nearest available station is returned.

Usage

```
select_METEO_stations(
  vWIN,
  interval = "daily",
  from_date = "2016-01-01",
  to_date = as.character(Sys.Date())
)
```

Arguments

`vWIN`	An ‘sf‘ or ‘sfc‘ spatial object defining the study area.
`interval`	Character string. Observation interval: ‘"minutely"‘, ‘"hourly"‘ or ‘"daily"‘.
`from_date`	Character string. First date of the requested period in ‘YYYY-MM-DD‘ format.
`to_date`	Character string. Last date of the requested period in ‘YYYY-MM-DD‘ format. Defaults to the current date.

Details

Station metadata are obtained through [load_stations_metadatas()]. The station catalogue is filtered to mainland departments up to 95, excludes station ‘17300848‘ and station type 5, and uses the ‘is_<interval>‘ field to check temporal-resolution availability.

Value

An ‘sf‘ object containing the selected meteorological station or stations, with point geometries in the station catalogue CRS.

See Also

[load_METEO()], [load_stations_metadatas()]

Examples

```
## Not run:
area <- sf::st_read("study_area.gpkg")
stations <- select_METEO_stations(area, interval = "daily",
                                  from_date = "2020-01-01")

## End(Not run)
```

Index

bytes_to_human, 3

calc_duration_size, 3

enableBookmarking(), *21*
export_wms_from_sf_fixed_res, 4

fct.read.meteofrance, 5
fit_model, 7
folder_size, 7

get_duration_size_models, 8
get_sources, 9

load_AWC, 9
load_BDTOPO, 10
load_CLC, 10
load_DEPTH, 11
load_ISBA, 11
load_LIDAR, 12
load_METEO, 12
load_MNE, 13
load_OCS, 14
load_ORTHO, 14
load_OSM, 15
load_PCRS, 15
load_RPG, 16
load_RPG_supplement, 16
load_SENTINEL, 17
load_SOL, 18
load_stations_metadatas, 19
load_TEXTURE, 20
load_UAV, 20

run_app, 21

seconds_to_human, 22
select_METEO_stations, 22