

Package: blackmarbler (via r-universe)

October 15, 2026

Title Black Marble Data and Statistics

Version 0.1.2

Description Geographically referenced data and statistics of nighttime lights from NASA Black Marble
<<https://blackmarble.gsfc.nasa.gov/>>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

URL <https://worldbank.github.io/blackmarbler/>

BugReports <https://github.com/worldbank/blackmarbler/issues>

Imports readr, hdf5r, dplyr, purrr, lubridate, tidyr, sf, curl,
exactextractr, stringr, httr2, terra, cli

Suggests geodata, ggplot2, knitr, leaflet, testthat (>= 3.0.0), fs

Depends R (>= 2.10)

LazyData true

Config/testthat/edition 3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libhdf5-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://dar4datascience.r-universe.dev>

Date/Publication 2024-05-04 00:53:17 UTC

RemoteUrl <https://github.com/dar4datascience/blackmarbler>

RemoteRef HEAD

RemoteSha 30a391cb22befeafe611002dada32d10c0402b51

Contents

add_n_pixels	2
apply_scaling_factor_to_viirs_data	3
bind_extracted_data	4
black_marble_tiles_sf	5
bm_extract	6
bm_raster	7
clean_raster_data	10
convert_h5_to_raster	11
count_n_obs	12
create_black_marble_dataset_df	12
create_raster_from_data_metadata	13
custom_inform	14
define_blackmarble_variable	14
define_raster_name	15
download_and_convert_raster	16
download_h5_viirs_sat_image	17
extract_and_process	18
extract_bounding_box	19
extract_daily_data	19
extract_data_and_metadata_from_hdf5	20
extract_monthly_data	21
intersect_bm_tiles	22
julian_to_month	23
map_black_marble_tiles	23
process_tiles	24
read_black_marble_csv	25
remove_fill_value_from_satellite_data	26
retrieve_and_process_nightlight_data	27
Index	29

add_n_pixels	<i>Add pixel information to extracted data</i>
--------------	--

Description

This function calculates additional pixel information for the extracted data frames.

Usage

```
add_n_pixels(roi_df, bm_r, roi_sf, quiet)
```

Arguments

roi_df	Data frame containing extracted data.
bm_r	The raster data.
roi_sf	The spatial features representing the regions of interest.
quiet	Logical indicating whether to show progress messages.

Value

The input data frame with added columns for pixel information.

Examples

```
## Not run:
# Adding pixel information to extracted data
add_n_pixels(roi_df, bm_r, roi_sf, quiet = FALSE)

## End(Not run)
```

apply_scaling_factor_to_viirs_data

Apply Scaling Factor to VIIRS Data

Description

Apply scaling factor to variables according to Black Marble user guide. The scaling factor is 0.1 for specific VIIRS variables.

Usage

```
apply_scaling_factor_to_viirs_data(x, variable, quiet = TRUE)
```

Arguments

x	A numeric vector or matrix representing the VIIRS data.
variable	A character string specifying the variable name.
quiet	Logical, if TRUE, suppresses warnings when the variable is not found in scaling variables.

Details

This function applies a scaling factor of 0.1 to specific VIIRS variables according to the Black Marble user guide.

The following VIIRS variables are affected:

- DNB_At_Sensor_Radiance (VNP46A1)
- DNB_BRDF-Corrected_NTL (VNP46A2)

- Gap_Filled_DNB_BRDF-Corrected_NTL (VNP46A2)
- DNB_Lunar_Irradiance (VNP46A2)
- AllAngle_Composite_Snow_Covered (VNP46A3/4)
- AllAngle_Composite_Snow_Covered_Std (VNP46A3/4)
- AllAngle_Composite_Snow_Free (VNP46A3/4)
- AllAngle_Composite_Snow_Free_Std (VNP46A3/4)
- NearNadir_Composite_Snow_Covered (VNP46A3/4)
- NearNadir_Composite_Snow_Covered_Std (VNP46A3/4)
- NearNadir_Composite_Snow_Free (VNP46A3/4)
- NearNadir_Composite_Snow_Free_Std (VNP46A3/4)
- OffNadir_Composite_Snow_Covered (VNP46A3/4)
- OffNadir_Composite_Snow_Covered_Std (VNP46A3/4)
- OffNadir_Composite_Snow_Free (VNP46A3/4)
- OffNadir_Composite_Snow_Free_Std (VNP46A3/4)

Value

A numeric vector or matrix with the scaling factor applied to the specified variables.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

bind_extracted_data	<i>Bind extracted data frames</i>
---------------------	-----------------------------------

Description

This function binds together the extracted data frames or a list of extracted data frames.

Usage

```
bind_extracted_data(extracted_data_list)
```

Arguments

extracted_data_list

One or more extracted data frames or a list of extracted data frames.

Value

A single data frame containing the bound extracted data.

Examples

```
## Not run:  
# Binding extracted data frames  
bind_extracted_data(df1, df2)  
  
# Binding a list of extracted data frames  
bind_extracted_data(list_of_dfs)  
  
## End(Not run)
```

black_marble_tiles_sf *Black Marble Tiles Spatial Data*

Description

Spatial dataset containing black marble tiles obtained from NASA's Black Marble project.

Usage

```
black_marble_tiles_sf
```

Format

A spatial dataframe (sf) with 648 rows and 2 columns:

TileID Identification of the tile associated with the VIIRS satellites mapping.

geometry Spatial geometry representing the tile boundaries.

Source

<https://blackmarble.gsfc.nasa.gov/>

The dataset is obtained from the Black Marble project by NASA, specifically the VIIRS (Visible Infrared Imaging Radiometer Suite) satellites. The TileID column represents the identification of the tile associated with the VIIRS satellites mapping. For more details on the dataset and its characteristics, refer to the Black Marble User Guide: https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

References

Tan, Bin, et al. "Night-time light imagery analysis for urban population mapping in China." *International Journal of Remote Sensing* 27.9 (2006): 1943-1950.

Wolfe, Robert E., et al. "Achieving accuracy requirements for forest biomass mapping: A data fusion method for combining the strengths of spaceborne lidar and Landsat TM data." *Remote Sensing of Environment* 80.2 (2002): 253-267.

See Also

https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

Examples

```
head(black_marble_tiles_sf)
```

bm_extract

Extract and process raster data for individual dates

Description

This function extracts and processes raster data for individual dates.

Usage

```
bm_extract(
  roi_sf,
  product_id,
  date,
  bearer,
  aggregation_fun = c("mean"),
  add_n_pixels = TRUE,
  variable = NULL,
  quality_flags_to_remove = NULL,
  check_all_tiles_exist = TRUE,
  interpol_na = FALSE,
  output_location_type = "memory",
  file_dir = NULL,
  file_prefix = NULL,
  file_skip_if_exists = TRUE,
  quiet = FALSE,
  ...
)
```

Arguments

roi_sf	The spatial features representing the regions of interest.
product_id	The product ID.
date	The date.
bearer	The bearer.
aggregation_fun	The function to aggregate data.
add_n_pixels	Logical indicating whether to compute additional pixel information.
variable	The variable.
quality_flags_to_remove	The quality flags to remove.
check_all_tiles_exist	Logical indicating whether to check if all tiles exist.

```

interpol_na    Logical indicating whether to interpolate missing values.
output_location_type
                The type of output location.
file_dir       The directory to save files.
file_prefix    The prefix for file names.
file_skip_if_exists
                Logical indicating whether to skip saving if files exist.
quiet          Logical indicating whether to suppress messages.
...           Additional arguments passed to helper functions.

```

Value

A data frame containing the extracted and processed raster data.

Examples

```

## Not run:
bm_extract(roi_sf = my_roi_sf, product_id = "my_product", date = "2024-04-02", bearer = "my_bearer")

## End(Not run)

```

bm_raster	<i>Extract and Aggregate Black Marble Data</i>
-----------	--

Description

Extract and aggregate nighttime lights data from NASA Black Marble data.

Usage

```

bm_raster(
  roi_sf,
  product_id,
  date,
  bearer,
  variable = NULL,
  quality_flags_to_remove = NULL,
  check_all_tiles_exist = TRUE,
  interpol_na = FALSE,
  output_location_type = "memory",
  file_dir = NULL,
  file_prefix = NULL,
  file_skip_if_exists = TRUE,
  quiet = FALSE,
  ...
)

```

Arguments

roi_sf	Region of interest; sf polygon. Must be in the WGS 84 (epsg:4326) coordinate reference system.
product_id	One of the following: <ul style="list-style-type: none"> • "VNP46A1": Daily (raw) • "VNP46A2": Daily (corrected) • "VNP46A3": Monthly • "VNP46A4": Annual
date	Date of raster data. Entering one date will produce a raster. Entering multiple dates will produce a raster stack. <ul style="list-style-type: none"> • For product_ids "VNP46A1" and "VNP46A2", a date (e.g., "2021-10-03"). • For product_id "VNP46A3", a date or year-month (e.g., "2021-10-01", where the day will be ignored, or "2021-10"). • For product_id "VNP46A4", year or date (e.g., "2021-10-01", where the month and day will be ignored, or 2021).
bearer	NASA bearer token. For instructions on how to create a token, see here .
variable	Variable to used to create raster (default: NULL). If NULL, uses the following default variables: <ul style="list-style-type: none"> • For product_id ":VNP46A1", uses DNB_At_Sensor_Radiance_500m. • For product_id "VNP46A2", uses Gap_Filled_DNB_BRDF-Corrected_NTL. • For product_ids "VNP46A3" and "VNP46A4", uses NearNadir_Composite_Snow_Free. For information on other variable choices, see here ; for VNP46A1, see Table 3; for VNP46A2 see Table 6; for VNP46A3 and VNP46A4, see Table 9.
quality_flags_to_remove	<p>Quality flag values to use to set values to NA. Each pixel has a quality flag value, where low quality values can be removed. Values are set to NA for each value in ther quality_flag_rm vector. (Default: NULL).</p> <p>For VNP46A1 and VNP46A2 (daily data):</p> <ul style="list-style-type: none"> • 0: High-quality, Persistent nighttime lights • 1: High-quality, Ephemeral nighttime Lights • 2: Poor-quality, Outlier, potential cloud contamination, or other issues <p>For VNP46A3 and VNP46A4 (monthly and annual data):</p> <ul style="list-style-type: none"> • 0: Good-quality, The number of observations used for the composite is larger than 3 • 1: Poor-quality, The number of observations used for the composite is less than or equal to 3 • 2: Gap filled NTL based on historical data
check_all_tiles_exist	Check whether all Black Marble nighttime light tiles exist for the region of interest. Sometimes not all tiles are available, so the full region of interest may not be covered. If TRUE, skips cases where not all tiles are available. (Default: TRUE).

interpol_na	When data for more than one date is downloaded, whether to interpolate NA values in rasters using the raster::approxNA function. Additional arguments for the raster::approxNA function can also be passed into bm_extract (eg, method, rule, f, ties, z, NA_rule). (Default: FALSE).
output_location_type	Where to produce output; either memory or file. If memory, functions returns a dataframe in R. If file, function exports a .csv file and returns NULL.
file_dir	(If output_location_type = file). The directory where data should be exported (default: NULL, so the working directory will be used)
file_prefix	(If output_location_type = file). Prefix to add to the file to be saved. The file will be saved as the following: <code>file_prefix_tdate.csv</code>
file_skip_if_exists	(If output_location_type = file). Whether the function should first check wither the file already exists, and to skip downloading or extracting data if the data for that date if the file already exists (default: TRUE).
quiet	Suppress output that show downloading progress and other messages. (Default: FALSE).
...	Additional arguments for raster::approxNA, if interpol_na = TRUE

Value

Raster

Examples

```
## Not run:
# Define bearer token
bearer <- "BEARER-TOKEN-HERE"

# sf polygon of Ghana
library(geodata)
roi_sf <- gadm(country = "GHA", level = 1, path = tempdir()) %>% st_as_sf()

# Daily data: raster for October 3, 2021
ken_20210205_r <- bm_raster(
  roi_sf = roi_sf,
  product_id = "VNP46A2",
  date = "2021-10-03",
  bearer = bearer
)

# Monthly data: raster for March 2021
ken_202103_r <- bm_raster(
  roi_sf = roi_sf,
  product_id = "VNP46A3",
  date = "2021-03-01",
  bearer = bearer
)
```

```
# Annual data: raster for 2021
ken_2021_r <- bm_raster(
  roi_sf = roi_sf,
  product_id = "VNP46A4",
  date = 2021,
  bearer = bearer
)

## End(Not run)
```

clean_raster_data	<i>Clean Raster Data</i>
-------------------	--------------------------

Description

This function cleans raster data by removing fill values and applying scaling factors.

Usage

```
clean_raster_data(raster_obj, variable_name)
```

Arguments

raster_obj	The raster object to be cleaned.
variable_name	A character string specifying the variable name.

Details

This function removes fill values and applies scaling factors to the specified variable in the raster object.

Value

A cleaned raster object.

See Also

[remove_fill_value_from_satellite_data](#), [apply_scaling_factor_to_viirs_data](#)

convert_h5_to_raster	<i>Convert HDF5 File to Raster</i>
----------------------	------------------------------------

Description

Converts an HDF5 file to a raster object.

Usage

```
convert_h5_to_raster(  
  download_path,  
  blackmarble_variable,  
  quality_flags_to_remove = numeric()  
)
```

Arguments

`download_path` A character string representing the download path to the HDF5 file.

`blackmarble_variable`
A character string specifying the blackmarble_variable name to extract from the HDF5 file.

`quality_flags_to_remove`
A numeric vector containing quality flag values to be removed from the data (optional).

Details

This function converts an HDF5 file to a raster object. It extracts the specified variable from the HDF5 file and optionally removes specific quality flag values from the data.

Value

A raster object containing the extracted variable data from the HDF5 file.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

count_n_obs	<i>Count Observations for Exact Extract</i>
-------------	---

Description

Counts observations for each variable, considering coverage fraction, for use in exact_extract.

Usage

```
count_n_obs(values)
```

Arguments

values	A data frame containing the values.
--------	-------------------------------------

Details

This function counts observations for each variable in the provided data frame, considering coverage fraction, for use in exact_extract. It calculates the number of non-NA pixels and total pixels for each variable. The coverage_fraction parameter is included in the function's signature but currently not used in the calculation.

Value

A data frame with the count of non-NA pixels and total pixels for each variable.

create_black_marble_dataset_df	<i>Create Black Marble Dataset DataFrame</i>
--------------------------------	--

Description

Creates a data frame containing Black Marble dataset filenames based on the specified parameters.

Usage

```
create_black_marble_dataset_df(
  product_id,
  all = TRUE,
  years = NULL,
  months = NULL,
  days = NULL
)
```

Arguments

product_id	The product ID specifying the type of Black Marble data.
all	Logical; indicating whether to create filenames for all available data or only for specific years, months, or days (default: TRUE).
years	A numeric vector specifying the years for which to create filenames (optional).
months	A numeric vector specifying the months for which to create filenames (optional).
days	A numeric vector specifying the days for which to create filenames (optional).

Details

This function generates a data frame with filenames of Black Marble datasets. It allows filtering by year, month, and day based on the provided parameters. Depending on the product_id, it generates filenames for daily, monthly, or yearly data. The generated filenames are based on the specified product ID, year, and day or month. The resulting data frame includes columns for the year and day or month, depending on the type of data.

Value

A data frame containing the filenames of Black Marble datasets based on the specified parameters.

Examples

```
## Not run:
# Generate filenames for all available Black Marble data
all_data <- create_black_marble_dataset_df("VNP46A1")

# Generate filenames for Black Marble data for specific years and months
specific_data <- create_black_marble_dataset_df("VNP46A2", years = c(2018, 2019), months = 1:6)

## End(Not run)
```

```
create_raster_from_data_metadata
```

Create Raster Object from Data and Metadata using terra

Description

This function creates a raster object from the provided data and metadata using the terra package.

Usage

```
create_raster_from_data_metadata(data, metadata)
```

Arguments

data	The data matrix for the raster.
metadata	A list containing metadata information including nodata value, bounding box coordinates, and coordinate reference system (CRS).

Details

This function transposes the input data, assigns nodata values to NA, creates an extent class from bounding box coordinates, and finally creates a raster object with the given data, extent, and CRS.

Value

A raster object created from the input data and metadata.

custom_inform	<i>Custom Inform Function</i>
---------------	-------------------------------

Description

This function prints an information message if quiet is set to FALSE.

Usage

```
custom_inform(message, quiet = FALSE)
```

Arguments

message	The message to be printed.
quiet	Logical indicating whether to suppress printing the message. Default is FALSE.

Examples

```
## Not run:
custom_inform("This is an information message.", quiet = FALSE)
custom_inform("This message will not be printed.", quiet = TRUE)

## End(Not run)
```

define_blackmarble_variable	<i>Define Black Marble Variable</i>
-----------------------------	-------------------------------------

Description

Defines the variable based on the Black Marble product ID if it is NULL.

Usage

```
define_blackmarble_variable(variable, product_id)
```

Arguments

variable	A character string specifying the variable to define.
product_id	A character string representing the product ID.

Details

This function defines the variable based on the provided Black Marble product ID if the variable is NULL. It checks if the product ID is valid and then assigns a default variable based on the product ID if the variable is NULL. The default variables for each product ID are as follows:

- VNP46A1: "DNB_At_Sensor_Radiance_500m"
- VNP46A2: "Gap_Filled_DNB_BRDF-Corrected_NTL"
- VNP46A3 and VNP46A4: "NearNadir_Composite_Snow_Free"

Value

A character string representing the defined variable.

define_raster_name	<i>Define Raster Name by Date</i>
--------------------	-----------------------------------

Description

Generates a name for the raster based on the given date and product ID.

Usage

```
define_raster_name(date_string, product_id)
```

Arguments

date_string	A character string representing the date in the format "YYYY-MM-DD".
product_id	A character string representing the product ID (e.g., "VNP46A1", "VNP46A2").

Details

This function generates a name for the raster based on the provided date and product ID. The raster name format depends on the product ID as follows:

- For VNP46A1: The raster name is simply the product ID.
- For VNP46A2: The raster name starts with "t" followed by the date string with hyphens replaced by underscores.
- For VNP46A3: The raster name starts with "t" followed by the year and month extracted from the date string.
- For VNP46A4: The raster name starts with "t" followed by the year extracted from the date string.

Value

A character string representing the generated raster name.

download_and_convert_raster

Download and Convert Raster Data

Description

Downloads raster data from NASA's LADSWeb and converts it to a raster object.

Usage

```
download_and_convert_raster(
  file_name,
  temp_dir,
  blackmarble_variable,
  bearer,
  quality_flags_to_remove = numeric(),
  quiet = FALSE
)
```

Arguments

file_name	A character string representing the name of the file to download.
temp_dir	A character string specifying the temporary directory where the file will be saved.
blackmarble_variable	A character string specifying the variable to extract from the raster data.
bearer	A character string containing the authorization token for accessing NASA's LADSWeb.
quality_flags_to_remove	A numeric vector containing quality flag values to be removed from the data (optional).
quiet	Logical; indicating whether to suppress progress messages (default: FALSE).

Details

This function downloads raster data from NASA's LADSWeb based on the provided file name. It then converts the downloaded data to a raster object, extracting the specified variable and removing quality flag values if specified. The function also provides an option to suppress progress messages.

Value

A raster object containing the downloaded and processed raster data.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

`download_h5_viirs_sat_image`*Download VIIRS Satellite Image in HDF5 Format*

Description

Downloads a VIIRS satellite image in HDF5 format from NASA's LADSWeb and saves it to a temporary directory or persistent location.

Usage

```
download_h5_viirs_sat_image(  
    file_name,  
    download_path,  
    bearer,  
    quality_flags_to_remove = numeric(),  
    quiet = FALSE  
)
```

Arguments

<code>file_name</code>	A character string representing the name of the file to download.
<code>download_path</code>	A character string representing the path to save the downloaded file.
<code>bearer</code>	A character string containing the authorization token for accessing NASA's LADSWeb.
<code>quality_flags_to_remove</code>	A numeric vector containing quality flag values to be removed from the data (optional).
<code>quiet</code>	Logical; indicating whether to suppress progress messages (default: FALSE).

Details

This function downloads a VIIRS satellite image in HDF5 format from NASA's LADSWeb based on the provided file name. It then processes the downloaded data, removing quality flag values if specified. The function also provides an option to suppress progress messages.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

extract_and_process	<i>Extract and process raster data</i>
---------------------	--

Description

This function extracts and processes raster data either for a single raster or for multiple rasters.

Usage

```
extract_and_process(
  bm_r,
  roi_sf,
  fun,
  add_n_pixels = TRUE,
  quiet = FALSE,
  is_single = FALSE
)
```

Arguments

bm_r	The raster data.
roi_sf	The spatial features representing the regions of interest.
fun	The function to apply to the raster data for extraction and processing.
add_n_pixels	Logical indicating whether to compute additional pixel information.
quiet	Logical indicating whether to show progress messages.
is_single	Logical indicating whether a single raster is being processed. Default is FALSE.

Details

This function extracts raster data from the specified regions of interest (roi_sf) and processes it using the specified function (fun). It can handle both single raster and multiple raster inputs.

Value

A data frame containing the extracted and processed raster data.

Examples

```
## Not run:
# For a single raster
extract_and_process(raster = my_raster, roi_sf = my_roi_sf, fun = mean, is_single = TRUE,
  add_n_pixels = TRUE, quiet = FALSE)

# For multiple rasters
extract_and_process(raster = my_raster, roi_sf = my_roi_sf, fun = mean)

## End(Not run)
```

extract_bounding_box	<i>Extracts bounding box coordinates based on tile ID from the file path</i>
----------------------	--

Description

This code snippet extracts the bounding box coordinates (minimum longitude, minimum latitude, maximum longitude, maximum latitude) based on the tile ID extracted from the file path. It uses the `file_path` to extract the tile ID, then retrieves the corresponding grid from the `black_marble_tiles_sf` dataset. Finally, it calculates the bounding box of the grid and rounds the coordinates.

Usage

```
extract_bounding_box(file_path, black_marble_tiles_sf)
```

Arguments

<code>file_path</code>	A character string specifying the file path.
<code>black_marble_tiles_sf</code>	The spatial dataset containing the grid tiles information.

Value

A list containing the minimum and maximum longitude and latitude coordinates of the bounding box.

extract_daily_data	<i>Extract Daily Data from HDF5 File</i>
--------------------	--

Description

This function extracts daily data from an HDF5 file based on the provided file path and variable name.

Usage

```
extract_daily_data(  
  download_file_path,  
  h5_data,  
  blackmarble_variable,  
  quality_flags_to_remove  
)
```

Arguments

download_file_path A character string specifying the file path.

h5_data The HDF5 file data.

blackmarble_variable A character string specifying the variable name.

quality_flags_to_remove A numeric vector containing quality flag values to be removed from the data.

Details

This function extracts daily data from an HDF5 file for the specified variable. It retrieves the data and applies quality flag removal if specified. Additionally, it calculates the bounding box coordinates based on the tile ID extracted from the file path.

Value

A list containing the extracted data and metadata.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

extract_data_and_metadata_from_hdf5

Extract Data and Metadata from HDF5 File

Description

This function extracts data and metadata from an HDF5 file.

Usage

```
extract_data_and_metadata_from_hdf5(
  h5_data,
  download_path,
  blackmarble_variable,
  quality_flags_to_remove
)
```

Arguments

h5_data The HDF5 file data.

download_path A character string specifying the path to the downloaded file.

blackmarble_variable A character string specifying the variable name.

quality_flags_to_remove A numeric vector containing quality flag values to be removed from the data.

Details

This function extracts data and metadata from an HDF5 file based on the specified file path and variable name. Depending on the file path pattern (daily or monthly/annually), it calls different functions to extract the data. Quality flag values specified in `quality_flags_to_remove` are removed from the data.

Value

A list containing the extracted data and metadata.

References

Black Marble User Guide - https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf

extract_monthly_data	<i>Extract Monthly Data from HDF5 File</i>
----------------------	--

Description

This function extracts monthly data from an HDF5 file.

Usage

```
extract_monthly_data(  
  download_file_path,  
  h5_data,  
  blackmarble_variable,  
  quality_flags_to_remove  
)
```

Arguments

download_file_path	Path to the downloaded HDF5 file.
h5_data	The HDF5 data.
blackmarble_variable	The variable to extract from the HDF5 data.
quality_flags_to_remove	Quality flag values to set to NA.

Value

A list containing the extracted data and bounding box coordinates.

Examples

```
## Not run:
# Extract monthly data
data <- extract_monthly_data(download_file_path, h5_data, blackmarble_variable, quality_flags_to_remove)

## End(Not run)
```

intersect_bm_tiles	<i>Intersect Black Marble Tiles with Region of Interest</i>
--------------------	---

Description

This function intersects Black Marble tiles with a region of interest, removing grid along edges to avoid issues with `st_intersects`.

Usage

```
intersect_bm_tiles(black_marble_tiles_sf, roi_sf)
```

Arguments

black_marble_tiles_sf	Spatial object representing Black Marble tiles.
roi_sf	Spatial object representing the region of interest.

Value

Spatial object containing Black Marble tiles intersecting with the region of interest.

Examples

```
## Not run:
# Load required libraries
library(sf)

# Create example data
black_marble_tiles_sf <- st_read("path_to_black_marble_tiles_shapefile")
roi_sf <- st_read("path_to_roi_shapefile")

# Intersect Black Marble tiles with region of interest
intersected_tiles <- intersect_bm_tiles(black_marble_tiles_sf, roi_sf)

## End(Not run)
```

julian_to_month*Translate Julian Dates to Regular Month Representation*

Description

This function translates Julian dates to regular month representations.

Usage

```
julian_to_month(julian_date)
```

Arguments

julian_date	A character string representing the day of the year in Julian format (e.g., "001" for January 1st).
-------------	---

Value

A number representing the month corresponding to the given Julian date (e.g., 1 for January).

References

For more information on the Julian day system, see: https://en.wikipedia.org/wiki/Julian_day

Examples

```
## Not run:  
julian_to_month("001")  
# [1] 1  
  
julian_to_month("032")  
# [1] 2  
  
## End(Not run)
```

map_black_marble_tiles*Map Black Marble Tiles*

Description

Generate an interactive map displaying black marble tiles using Leaflet.

Usage

```
map_black_marble_tiles()
```

Details

This function requires the leaflet package for generating interactive maps.

This function generates an interactive map displaying black marble tiles obtained from NASA's Black Marble project using the Leaflet library. The map includes black marble tiles overlaid on a base map, with a marker indicating the center coordinates (latitude and longitude).

Value

An interactive Leaflet map object.

Examples

```
## Not run:
library(leaflet)
# Generate the map
map_black_marble_tiles()
```

```
## End(Not run)
```

process_tiles

Process Tiles

Description

Processes satellite imagery tiles based on specified parameters.

Usage

```
process_tiles(
  bm_files_df,
  grid_use_sf,
  check_all_tiles_exist,
  temp_dir,
  product_id,
  blackmarble_variable,
  bearer,
  quality_flags_to_remove,
  quiet
)
```

Arguments

bm_files_df A data frame containing Black Marble dataset filenames.

grid_use_sf A spatial grid containing TileIDs.

check_all_tiles_exist Logical; indicating whether to check if all tiles exist (default: TRUE).

temp_dir	A character string specifying the temporary directory.
product_id	A character string representing the product ID.
blackmarble_variable	A character string specifying the Black Marble variable.
bearer	A character string containing the authorization token for accessing NASA's LADSWeb.
quality_flags_to_remove	A numeric vector containing quality flag values to be removed from the data (optional).
quiet	Logical; indicating whether to suppress progress messages (default: FALSE).

Value

A raster object processed from satellite imagery tiles.

read_black_marble_csv *Read Black Marble CSV Data*

Description

Reads Black Marble CSV data for a specific year and day.

Usage

```
read_black_marble_csv(year, day, product_id)
```

Arguments

year	The year of the data.
day	The day of the year (1-365 or 1-366 for leap years).
product_id	The product ID specifying the type of data to retrieve.

Details

This function reads Black Marble CSV data from the NASA LADS website for the given year, day, and product_id. It constructs the URL based on the provided parameters and attempts to read the CSV file using `readr::read_csv`. If successful, it adds columns for year and day to the data frame. If an error occurs during the reading process, it returns an empty data frame and issues a warning. Additionally, to avoid overloading the server with rapid requests, the function includes a small delay (0.1 seconds) between consecutive requests using `Sys.sleep(0.1)`.

Value

A data frame containing the Black Marble CSV data for the specified year and day.

Examples

```
## Not run:
# Read Black Marble CSV data for the year 2023, day 150, and product ID "VIIRS_SNPP_CorrectedReflectance_BandM3"
data <- read_black_marble_csv(2023, 150, "VIIRS_SNPP_CorrectedReflectance_BandM3")

## End(Not run)
```

```
remove_fill_value_from_satellite_data
```

Remove Artifact Values from Satellite Data

Description

This function removes artifact values from satellite data, replacing them with NA. Artifact values are commonly used in satellite data to represent fill values, invalid measurements, or missing data.

Usage

```
remove_fill_value_from_satellite_data(data, blackmarble_variable)
```

Arguments

data	A matrix or data frame containing satellite data.
blackmarble_variable	A character string specifying the blackmarble_variable for which artifact values should be removed.

Details

The artifact values and their corresponding blackmarble_variables are based on the data format described in the Black Marble User Guide (https://viirsland.gsfc.nasa.gov/PDF/BlackMarbleUserGuide_v1.2_20220916.pdf).

Value

The input data with artifact values replaced by NA.

*** Table 3 (page 12)**

- Table 6 (page 16)
- Table 9 (page 18)

References

For more information on artifact values in satellite data, refer to the Black Marble User Guide.

Examples

```
## Not run:
# Remove artifact values from UTC_Time variable in satellite data
cleaned_data <- remove_fill_value_from_satellite_data(satellite_data, "UTC_Time")

## End(Not run)
```

```
retrieve_and_process_nightlight_data
```

```
Retrieve and Process Nightlight Data
```

Description

Retrieves and processes nighttime light raster data from the Black Marble dataset.

Usage

```
retrieve_and_process_nightlight_data(
  roi_sf,
  product_id,
  date,
  bearer,
  blackmarble_variable,
  quality_flags_to_remove,
  check_all_tiles_exist = TRUE,
  quiet = FALSE,
  temp_dir
)
```

Arguments

<code>roi_sf</code>	An sf object representing the region of interest.
<code>product_id</code>	A character string specifying the product ID.
<code>date</code>	A character string representing the date.
<code>bearer</code>	A character string representing the authorization bearer token.
<code>blackmarble_variable</code>	A character string specifying the blackmarble_variable to extract.
<code>quality_flags_to_remove</code>	A numeric vector containing quality flag values to be removed from the data (optional).
<code>check_all_tiles_exist</code>	A logical value indicating whether to check if all satellite imagery tiles for the location exist (default is TRUE).
<code>quiet</code>	A logical value indicating whether to suppress progress messages (default is FALSE).
<code>temp_dir</code>	A character string representing the temporary directory path.

Details

This function retrieves and processes nighttime light raster data from the Black Marble dataset. It downloads the data for the specified region of interest and date, removes any quality-flagged pixels as specified, and mosaics the tiles together if necessary. It then crops the raster to the region of interest.

Value

A raster object containing the processed nighttime light data.

Index

- * **datasets**
 - black_marble_tiles_sf, [5](#)
- add_n_pixels, [2](#)
- apply_scaling_factor_to_viirs_data, [3](#),
[10](#)
- bind_extracted_data, [4](#)
- black_marble_tiles_sf, [5](#)
- bm_extract, [6](#)
- bm_raster, [7](#)
- clean_raster_data, [10](#)
- convert_h5_to_raster, [11](#)
- count_n_obs, [12](#)
- create_black_marble_dataset_df, [12](#)
- create_raster_from_data_metadata, [13](#)
- custom_inform, [14](#)
- date, [9](#)
- define_blackmarble_variable, [14](#)
- define_raster_name, [15](#)
- download_and_convert_raster, [16](#)
- download_h5_viirs_sat_image, [17](#)
- extract_and_process, [18](#)
- extract_bounding_box, [19](#)
- extract_daily_data, [19](#)
- extract_data_and_metadata_from_hdf5,
[20](#)
- extract_monthly_data, [21](#)
- file_prefix, [9](#)
- intersect_bm_tiles, [22](#)
- julian_to_month, [23](#)
- map_black_marble_tiles, [23](#)
- process_tiles, [24](#)
- read_black_marble_csv, [25](#)
- remove_fill_value_from_satellite_data,
[10](#), [26](#)
- retrieve_and_process_nightlight_data,
[27](#)