Package: surfaceuv (via r-universe)

October 31, 2024

Type Package

Title Solar Ultraviolet at Ground Surface Data Import

Version 0.1.0.9000

Date 2024-10-27

Description Functions for reading files containing data on ``Surface UV'' such as daily doses, irradiances and theUV Index (UVI). Curently the Surface UV product from the AC SAF project of EUMETSAT is supported in both geographic grid (HDF5) and time series (text) formats.

License GPL (>= 2)

VignetteBuilder knitr

Depends R (>= 4.0.0)

Imports utils, bit64, rhdf5 (>= 2.45.0)

Suggests knitr (>= 1.43), rmarkdown (>= 2.23), spelling, testthat (>= 3.1.10)

LazyLoad yes

ByteCompile true

Encoding UTF-8

URL https://docs.r4photobiology.info/surfaceuv/

BugReports https://github.com/aphalo/surfaceuv/issues/

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Language en-US

Config/testthat/edition 3

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

RemoteUrl https://github.com/aphalo/surfaceuv

RemoteRef HEAD

RemoteSha b59933b58067b618fc5750668f7c5f7cea5bfb89

Contents

`read_AC_SAF_UV_hdf5` *Offline AC SAF gridded Surface UV*

Description

Import **gridded** "Surface UV" data released by EUMETSAT AC SAF (Atmospheric Composition Monitoring) project from **HDF5** files downloaded from the FMI server.

Usage

```
read_AC_SAF_UV_hdf5(
  files,
  data.product = NULL,
  group.name = "GRID_PRODUCT",
  vars.to.read = NULL,
  fill = NA_real_,
  keep.QC = TRUE,
  verbose = interactive()
)

vars_AC_SAF_UV_hdf5(
  files,
  data.product = NULL,
  group.name = "GRID_PRODUCT",
  keep.QC = TRUE,
  set.oper = "intersect"
)

grid_AC_SAF_UV_hdf5(files, expand = FALSE)

date_AC_SAF_UV_hdf5(files, use.names = length(files > 1))
```

Arguments

`files`	character A vector of file names, no other limitation in length than available memory to hold the data.
`data.product`	character Currently only "Surface UV" supported.
`group.name`	character The name of the 'group' in the HDF5 files, or a regular expression for matching a single group name with `grep()`.
`vars.to.read`	character A vector of variable names. If `NULL` all the variables present in the first file are read.

`fill` numeric The R value used to replace the fill value used in the file, which is retrieved from the file metadata, and also used to fill missing variables.

`keep.QC` logical Add to the returned data frame or vector the quality control variable, always present in the files.

`verbose` logical Flag indicating if progress, and time and size of the returned object should be printed.

`set.oper` character One of `"intersect"`, or `"union"`.

`expand` logical Flag indicating whether to return ranges or a full grid.

`use.names` logical. Should names be added to the returned vector?

Details

Function `read_AC_SAF_UV_hdf5()` can be used to read the data stored in a file, either in full or selected variables. Query functions `vars_AC_SAF_UV_hdf5()`, `grid_AC_SAF_UV_hdf5()` and `date_AC_SAF_UV_hdf5()` extract the names of the variables, the range of the grid and the dates of measurements much more efficiently than by using `read_AC_SAF_UV_hdf5()`. The dates are decoded from the file names, expecting these to be those set by the data provider. The grid is expected to be identical in all files that are imported in a call to `read_AC_SAF_UV_hdf5()`, and grid subsetting is currently not supported. If not all the files named in the argument to `files` are accessible, an error is triggered early. If the files differ in the grid, an error is triggered when reading the first mismatching file. Missing variables named in `vars.to.read` if detected when reading the first file, are filled with the `fill` value, otherwise they trigger an error when an attempt is made to read them.

Value

Function `read_AC_SAF_UV_hdf5()` returns a data frame with columns named `"Date"`, `"Longitude"`, `"Latitude"`, the data variables with their original names, and `"QualityFlags"`. The data variables have their metadata stored as R attributes. `vars_AC_SAF_UV_hdf5()` returns a `character` vector of variable names, `grid_AC_SAF_UV_hdf5()` returns a data frame with two numeric variables, `Longitude` and `Latitude`, with two rows or an expanded grid depending on the argument passed to `expand`, while `date_AC_SAF_UV_hdf5()` returns a named vector of class `Date`, with file names as names.

Note

The constraint on the consistency among all files to be read allows very fast reading into a single data frame. If the files differ in the grid or set of variables, this function can be used to read the files individually into separate data frames. These data frames can later be row-bound together.

Variable `QualityFlags` is encoded as 64 bit integers in the HDF5 file and read as a double. R package 'bit64' can be used to print these values as 64 bit integers.

When requesting the data from the EUMETSAT AC SAF FMI server at `https://acsaf.org/` it is possible to select the range of latitudes and longitudes and the variables to be included in the file. This is more efficient than doing the selection after importing the data into R. The data are returned as a .zip compressed file containing one .HDF5 file for each day in the range of dates selected. For world coverage each of these files can be as large as 10 MB in size depending on how many variables they contain. These files in HDF5 format are binary files so the size in RAM of a `data.frame` object containing one-year of data can be a few 10's of GB.

This function's performance is fast as long as there is enough RAM available to hold the data frame and the files are read from a reasonably fast SSD. The example data included in the package are only for Spain and five summer days. They are used in examples and automated tests. Function `read_AC_SAF_UV_hdf5()` has been also tested by importing one-year's worth of data with world-wide coverage on a PC with 64GB RAM.

References

Kujanpää, J. (2019) *PRODUCT USER MANUAL Offline UV Products v2 (IDs: O3M-450 - O3M-464) and Data Record R1 (IDs: O3M-138 - O3M-152).* Ref. SAF/AC/FMI/PUM/001. 18 pp. EUMETSAT AC SAF.

See Also

`read_AC_SAF_UV_txt()` supporting the same Surface UV data stored in text files as single-location time series.

Examples

```
# find location of one example file
one.file.name <-
   system.file("extdata", "O3MOUV_L3_20240621_v02p02.HDF5",
               package = "surfaceuv", mustWork = TRUE)

# available variables
vars_AC_SAF_UV_hdf5(one.file.name)

# available grid
grid_AC_SAF_UV_hdf5(one.file.name)

# decode date from file name
date_AC_SAF_UV_hdf5(one.file.name)
date_AC_SAF_UV_hdf5(one.file.name, use.names = FALSE)

# read all variables
midsummer_spain.tb <- read_AC_SAF_UV_hdf5(one.file.name)
dim(midsummer_spain.tb)
summary(midsummer_spain.tb)

# read two variables
midsummer_spain_daily.tb <-
  read_AC_SAF_UV_hdf5(one.file.name,
                      vars.to.read = c("DailyDoseUva", "DailyDoseUvb"))
dim(midsummer_spain_daily.tb)
summary(midsummer_spain_daily.tb)

# find location of three example files
three.file.names <-
   system.file("extdata",
               c("O3MOUV_L3_20240621_v02p02.HDF5",
                 "O3MOUV_L3_20240622_v02p02.HDF5",
                 "O3MOUV_L3_20240623_v02p02.HDF5"),
                package = "surfaceuv", mustWork = TRUE)

date_AC_SAF_UV_hdf5(three.file.names)

summer_3days_spain.tb <- read_AC_SAF_UV_hdf5(three.file.names)
dim(summer_3days_spain.tb)
summary(summer_3days_spain.tb)
```

read_AC_SAF_UV_txt *Offline AC SAF Surface UV time series*

Description

Import **time series** "Surface UV" data released by EUMETSAT AC SAF (Atmospheric Composition Monitoring) project from **text** files downloaded from the FMI server.

Usage

```
read_AC_SAF_UV_txt(
  files,
  vars.to.read = NULL,
  add.geo = length(files) > 1,
  keep.QC = TRUE,
  verbose = interactive()
)

vars_AC_SAF_UV_txt(files, keep.QC = TRUE, set.oper = "intersect")

grid_AC_SAF_UV_txt(files, use.names = length(files) > 1)
```

Arguments

`files` character A vector of file names, no other limitation in length than available memory to hold the data.

`vars.to.read` character A vector of variable names. If `NULL` all the variables present in the first file are read.

`add.geo` logical Add columns `Longitude` and `Latitude` to returned data frame.

`keep.QC` logical Add to the returned data frame or vector the quality control variables, always present in the files.

`verbose` logical Flag indicating if progress, and time and size of the returned object should be printed.

`set.oper` character One of `"intersect"`, or `"union"`.

`use.names` logical. Should row names be added to the returned data frame?

Details

All information is in the files, including dates, and no information is decoded from file names, that users will most likely want to rename. Each file corresponds to a single geographic location. If not all the files named in the argument to `files` are accessible, an error is triggered early. If the files differ in the coordinates, an error is triggered when reading the first mismatching file if coordinates are not being added to the data frame. Missing variables named in `vars.to.read` are currently ignored.

Data from multiple files are concatenated. By default, the geographic coordinates are added in such a case.

Value

`read_AC_SAF_UV_txt()` returns a data frame with columns named `"Date"`, `"Longitude"`, `"Latitude"`, and the data variables with their original names (with no units). The data variables have no metadata stored as R attributes. When reading multiple files, by default the format is similar to that from function `read_AC_SAF_UV_hdf5()`. Column names are the same but column order can differ. File headers are saved as a list in R attribute `file.headers`. `vars_AC_SAF_UV_txt()` returns a `character` vector of variable names, and `grid_AC_SAF_UV_txt()` a dataframe with two numeric variables, `Longitude` and `Latitude`, and a single row.

Note

When requesting the data from the EUMETSAT AC SAF FMI server at https://acsaf.org/ it is possible to select the variables to be included in the file, the period and the geographic coordinates of a single location. The data are returned as a .zip compressed file containing one text file with one row for each day in the range of dates selected. These files are fairly small.

This function's performance is not optimized for speed as these single location files are rather small. The example time series data included in the package are for one summer in Helsinki, Finland.

References

Kujanpää, J. (2019) *PRODUCT USER MANUAL Offline UV Products v2 (IDs: O3M-450 - O3M-464) and Data Record R1 (IDs: O3M-138 - O3M-152)*. Ref. SAF/AC/FMI/PUM/001. 18 pp. EUMETSAT AC SAF.

See Also

`read_AC_SAF_UV_hdf5()` supporting the same Surface UV data stored in a gridded format.

Examples

```
# find location of one example file
one.file.name <-
   system.file("extdata", "AC_SAF-Viikki-FI-6masl.txt",
               package = "surfaceuv", mustWork = TRUE)

# Available variables
vars_AC_SAF_UV_txt(one.file.name)
vars_AC_SAF_UV_txt(one.file.name, keep.QC = FALSE)
```

```
# Grid point coordinates
grid_AC_SAF_UV_txt(one.file.name)

# read all variables
summer_viikki.tb <-
  read_AC_SAF_UV_txt(one.file.name)
dim(summer_viikki.tb)
colnames(summer_viikki.tb)
str(sapply(summer_viikki.tb, class))
summary(summer_viikki.tb)
attr(summer_viikki.tb, "file.headers")

# read all data variables
summer_viikki_QCf.tb <-
  read_AC_SAF_UV_txt(one.file.name, keep.QC = FALSE)
dim(summer_viikki_QCf.tb)
summary(summer_viikki_QCf.tb)

# read all data variables including geographic coordinates
summer_viikki_geo.tb <-
  read_AC_SAF_UV_txt(one.file.name, keep.QC = FALSE, add.geo = TRUE)
dim(summer_viikki_geo.tb)
summary(summer_viikki_geo.tb)

# read two variables
summer_viikki_2.tb <-
  read_AC_SAF_UV_txt(one.file.name,
                     vars.to.read = c("DailyDoseUva", "DailyDoseUvb"))
dim(summer_viikki_2.tb)
summary(summer_viikki_2.tb)
```

Index