Package 'CopernicusDEM'

August 17, 2024

Type Package

Title Copernicus Digital Elevation Models

Version 1.0.4

Date 2024-08-17

URL https://github.com/mlampros/CopernicusDEM

Description Copernicus Digital Elevation Model datasets (DEM) of 90 and 30 meters resolution using the 'awscli' command line tool. The Copernicus (DEM) is included in the Registry of Open Data on 'AWS (Amazon Web Services)' and represents the surface of the Earth including buildings, infrastructure and vegetation.

License GPL-3

Copyright inst/COPYRIGHTS.pdf

Encoding UTF-8

SystemRequirements awscli: apt install -y awscli (deb), aws-configure: aws configure (deb)

Depends R(>= 3.5)

Imports glue, sf, utils, doParallel, foreach

RoxygenNote 7.3.0

Suggests rmarkdown, knitr, data.table, fitbitViz, tmap, leaflet, leafgl

VignetteBuilder knitr

NeedsCompilation no

Author Lampros Mouselimis [aut, cre] (<https://orcid.org/0000-0002-8024-1546>)

Maintainer Lampros Mouselimis <mouselimislampros@gmail.com>

Repository CRAN

Date/Publication 2024-08-17 05:10:06 UTC

Contents

---

aoi_geom_save_tif_matches

> *Download the elevation .tif files that intersect either with an input sf*
> *(simple features) object or with a .geojson file*

Description

Download the elevation .tif files that intersect either with an input sf (simple features) object or with a .geojson file

Usage

```
aoi_geom_save_tif_matches(
  sf_or_file,
  dir_save_tifs,
  resolution = 90,
  crs_value = 4326,
  threads = parallel::detectCores(),
  verbose = FALSE
)
```

Arguments

sf_or_file	either an 'sf'(simple features) object or a .geojson file specifying the AOI (Area of Interest) for which the Digital Elevation Models (DEM) files should be downloaded
dir_save_tifs	a valid path to a directory where the .tif files should be saved
resolution	an integer value specifying the elevation resolution. The Copernicus Digital ELevation Models (DEM) currently include 90 and 30 meter resolution data
crs_value	an integer value specifying the Coordinates Reference System (CRS) value of the Digital ELevation Models (DEM) which is by default 4326
threads	an integer that specifies the number of threads to use in parallel when downloading the .tif files
verbose	a boolean. If TRUE then information will be printed out in the console

Details

Download Computation time: Based on a sample of 90 meter resolution images that I downloaded each file was approximately 5 MB which means in total I had to download 130 GB of data (in case I intended to download all 20.000 files of the land areas worldwide). Therefore it is wise to download data based on the intersection of the input Area of Interest (AOI) and an existing tile-grid of the Digital Elevation Model (DEM)

The 30 meter resolution .tif images are bigger in size but visually better (approximate image size of 1.7 MB compared to 13 MB). The time to download 90 meter resolution data is approximately 20 seconds compared to 1 minute and 10 seconds of the 30 meter resolution data (for a sample use case)

Value

a list object of length 2

References

https://registry.opendata.aws/copernicus-dem/

https://copernicus-dem-30m.s3.amazonaws.com/readme.html

https://spacedata.copernicus.eu/en/web/guest/collections/copernicus-digital-elevation-model/

Examples

```
## Not run:

#......................................
# create a directory to save the .tif
# files based on a Well Known Text (WKT)
# of a sample Area of Interest (AOI)
#......................................

DIR_SAVE = file.path(Sys.getenv('HOME'), 'DIR_SAVE_DEM')
if (!dir.exists(DIR_SAVE)) dir.create(DIR_SAVE)

WKT='POLYGON((61.5234 27.0591, 63.6328 27.0591, 63.6328 28.1495, 61.5234 28.1495, 61.5234 27.0591))'

sf_obj = sf::st_as_sfc(WKT, crs = 4326)
sf_obj = sf::st_make_valid(sf_obj)

#.............
# 90 meter DEM
#.............

save_matches = CopernicusDEM::aoi_geom_save_tif_matches(sf_or_file = sf_obj,
                                                        dir_save_tifs = DIR_SAVE,
                                                        resolution = 90,
                                                        crs_value = 4326,
                                                       threads = parallel::detectCores(),
                                                        verbose = TRUE)

#.............
# 30 meter DEM
#.............

save_matches = CopernicusDEM::aoi_geom_save_tif_matches(sf_or_file = sf_obj,
                                                        dir_save_tifs = DIR_SAVE,
                                                        resolution = 30,
                                                        crs_value = 4326,
                                                       threads = parallel::detectCores(),
                                                        verbose = TRUE)

## End(Not run)
```

---

create_VRT_from_dir   *Create a Virtual Raster (VRT) file from .tif files*

Description

Create a Virtual Raster (VRT) file from .tif files

Usage

```
create_VRT_from_dir(
  dir_tifs,
  output_path_VRT,
  file_extension = ".tif",
  verbose = FALSE
)
```

Arguments

dir_tifs	a valid path to a directory where the .tif files are saved
output_path_VRT	
	a valid path to a file where the Virtual Raster (VRT) will be saved
file_extension	a character string specifying the image file extension from which the .vrt file will be built
verbose	a boolean. If TRUE then information will be printed out in the console

Value

it doesn't return an object but it saves the output to a file

Examples

```
## Not run:

#.........................................................
# create a directory to save the .tif files and a
# Well Known Text (WKT) of a sample Area of Interest (AOI)
#.........................................................

DIR_SAVE = file.path(Sys.getenv('HOME'), 'DIR_SAVE_DEM')
if (!dir.exists(DIR_SAVE)) dir.create(DIR_SAVE)

WKT='POLYGON((61.5234 27.0591, 63.6328 27.0591, 63.6328 28.1495, 61.5234 28.1495, 61.5234 27.0591))'

sf_obj = sf::st_as_sfc(WKT, crs = 4326)
sf_obj = sf::st_make_valid(sf_obj)

#....................
# download 90 meter DEM
```

```
#.....................

save_matches = CopernicusDEM::aoi_geom_save_tif_matches(sf_or_file = sf_obj,
                                                        dir_save_tifs = DIR_SAVE,
                                                        resolution = 90,
                                                        crs_value = 4326,
                                                      threads = parallel::detectCores(),
                                                        verbose = TRUE)

#......................................
# create a Virtual Raster (VRT) file from
# the 90 meter downloaded .tif files
#......................................

VRT_out = as.character(glue::glue("{DIR_SAVE}.vrt"))

res_vrt = CopernicusDEM::create_VRT_from_dir(dir_tifs = DIR_SAVE,
                                             output_path_VRT = VRT_out,
                                             verbose = TRUE)

#....................................................
# load the saved VRT file as raster (which might
# consist of multiple files, i.e. a mosaic) and plot it
#....................................................

rst = raster::raster(VRT_out)
sp::plot(rst)


## End(Not run)
```

Index