

# Acquiring EO datasets for eLTER+ sites

true true

20/02/2021

## Contents

Setup	1
MODIS products, layers	3
Time series EO products averaged for each site	5
Visualization	5

## Setup

Load necessary R libraries, user configurable directories, then read in the `functions.R` script with contains helper functions for summarizing layers by date and site, and plotting graphs.

## Libraries

```
pkg_list = c("MODISTsp", "lubridate", "raster", "ggplot2",
             "cowplot", "tidyr", "sf", "stars", "leaflet",
             "shiny", "shinydashboard", "shinyFiles",
             "shinyalert", "rappdirs", "shinyjs", "leafem",
             "mapedit", "magrittr")
installed_packages <- pkg_list %in% rownames(installed.packages())
if (any(installed_packages == FALSE)) {
  install.packages(pkg_list[!installed_packages], dependencies = TRUE)
}
# Packages loading
pkgs = lapply(pkg_list, library, character.only = TRUE)
```

## Define directories

This code chunk includes reading a text file “site\_shapefiles\_url.txt” that includes a list of sites, with three columns: name, full\_name, url The URL is a link to the boundary shapefile from DEIMS for each site.

```
# Edit below as necessary: GIS, output, and figures directories
# and read options and sites files
GIS_dir = "../GIS"
if (!dir.exists(GIS_dir)) {dir.create(GIS_dir,
                                     recursive = TRUE)}

# Where to save outputs
Output_dir = "../Output"
if (!dir.exists(Output_dir)) {dir.create(Output_dir,
                                         recursive = TRUE)}
```

```

# Where to save figures
Figures_dir = "../Figures"
if (!dir.exists(Figures_dir)) {dir.create(Figures_dir,
                                         recursive = TRUE)}

# List of eLTER+ sites and DEIMS URL's for download
site_list_file = "site_shapefiles_url.txt"
site_list = read.csv(site_list_file)
sites = site_list$site_name

# (Add option to ignore Datum unknown warnings)
options("rgdal_show_exportToProj4_warnings"="none")

# Load some helper functions
source("functions.R")

```

### Load polygons from DEIMS site

Download shapefiles from list of eLTER sites Save each as geopackage The list of sites and download URL is in: "site\_shapefiles\_url.txt"

```

# Call ObtainSitePolygons function (in functions.R)
ObtainSitePolygons(site_list_file)

```

## MODIS products, layers

Use the MODISrsp package (Busetto and Ranghetti (2016)) to filter and download layers.

Display lists of all available products and layers in each product category.

```
MODISrsp_get_prodnames()
# [105] "Vegetation Indices_16Days_250m (M*D13Q1)"
# [23] "LST_3band_emissivity_8day_1km (M*D21A2)"
# ... many more

MODISrsp_get_prodlayers("Vegetation Indices_16Days_250m (M*D13Q1)")
MODISrsp_get_prodlayers("LST_3band_emissivity_8day_1km (M*D21A2)")
```

## Use the GUI

Here the user can choose:

- product, layers
- start and end dates
- a polygon area of interest (shapefile or Geopackage)
- and satellite platforms
- Each set of options saved to \*.json file

Requires registration on EarthData website: <https://urs.earthdata.nasa.gov/home>

Example: In “**Products and Layers**” panel

- from Product Category dropdown
  - choose Ecosystem variables Vegetation Indices
- from Product Name dropdown
  - choose Vegetation\_Indices\_16days\_250m
- from layers to be processed dropdown
  - choose 16 day NDVI average
- from Platform
  - choose Both

In “**Spatial Temporal**” panel

- in Temporal Range, select date range
- in Output Projection
  - select User defined
  - click “Change” and enter EPSG for desired projection
  - i.e. 3035 for ETRS89 based European LAEA (conformal) projection
- in Spatial Extent choose “Load from Spatial file” and click browse to choose gpkg for site

In **Output Format**

- Under Download Method, enter username and password
- Under Output Options, choose R rasterStack
- Under Output Folders, click browse to select output location

Click **Save Options**

- Save as json file
- Browse to save under R code directory

```
MODISrsp()
```

## Loop over all sites

Call the MODISTsp() function with `gui = FALSE` and point to each json formatted options file to run the download. The options file was saved from the GUI step above. This loop downloads all available MODIS tiles for each AOI.

The download utility used here is “aria2”. It can be obtained from: <https://github.com/aria2/aria2/releases/tag/release-1.35.0>

You **must** supply a username and password for authentication on the EarthData website

This code block will run for a **long** time.

```
#-----  
# Enter username and password here for EarthData website  
user = 'your user name'  
password = 'your password'  
#-----  
config_files = list.files(".", pattern = ".json$",  
                           full.names = TRUE)  
spatial_files = list.files(GIS_dir, pattern = ".gpkg$",  
                           full.names = TRUE)  
  
# Loop over sites  
lapply(spatial_files, FUN = function(site) {  
  t0 = Sys.time()  
  site_name = basename(tools::file_path_sans_ext(site))  
  print(paste(t0, "-- Processing site:", site_name))  
  # Loop over configurations  
  lapply(config_files, FUN = function(cfg) {  
    MODISTsp(gui = FALSE,  
             opts_file = cfg,  
             spafile = site,  
             spameth = "file",  
             user = user,  
             password = password,  
             #start_date = "2018.10.01", # To change the dates  
             #sensor = "Aqua", # "Terra" or "Both"  
             downloader = "aria2", # "html" or "aria2" if it is installed  
             verbose = FALSE  
          )  
  })  
  t1 = Sys.time()  
  elapsed = round(difftime(t1, t0, units = "mins"))  
  print(paste(t0, "-- Completed site:", site_name,  
             "in", elapsed, "mins"))  
})
```

## Time series EO products averaged for each site

Loop over all sites and summarize pixels by date for each site. The functions used here are stored in `functions.R`

### Site timeseries data

```
# Call TimeSeriesFromRaster() function for each site
# Create graphs of each time series with PlotTimeSeries() function
for (site in sites){
  t0 = Sys.time()
  print(paste(t0, "-- Time series for site:", site))
  timeseries_list = TimeSeriesFromRasters(site)
  PlotTimeSeries(timeseries_list, site)
}
```

### Corine Landcover for four years of CLC rasters

Corine Landcover rasters at 100 m resolution, for four years. Have been downloaded in advance from: <https://land.copernicus.eu/pan-european/corine-land-cover> Crop each raster to extent of the site bounding box

```
# Crop Corine Landcover from four years for each site
# Call CropSaveCorine() function for each site

for (site in sites) {
  CropSaveCorine(site)
}
```

## Visualization

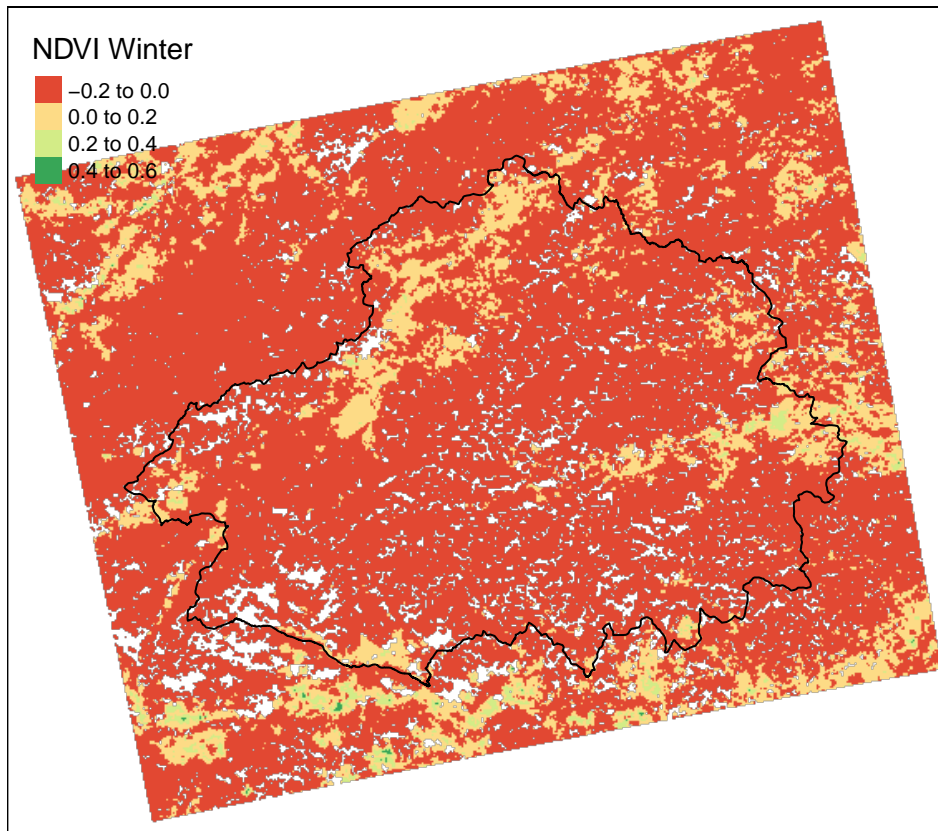
Show two NDVI maps and an example time series plot from Cairngorms

```
library("tmap")
library("tmtools") # For static maps
site = "Cairngorms"
site_gpkg = file.path(GIS_dir, paste0(site, ".gpkg"))
site_sf = sf::read_sf(site_gpkg)
site_sf = st_transform(site_sf, 4326)
NDVI_file_list = list.files(file.path(Output_dir,
                                       site,
                                       "VI_16Days_250m_v6/NDVI"),
                             pattern = ".tif$",
                             full.names = TRUE)
NDVI_1 = projectRaster(raster(NDVI_file_list[[1]]), crs = 4326)
NDVI_1 = NDVI_1 * 0.0001
NDVI_2 = projectRaster(raster(NDVI_file_list[[200]]), crs = 4326)
NDVI_2 = NDVI_2 * 0.0001
tmap_mode("plot")

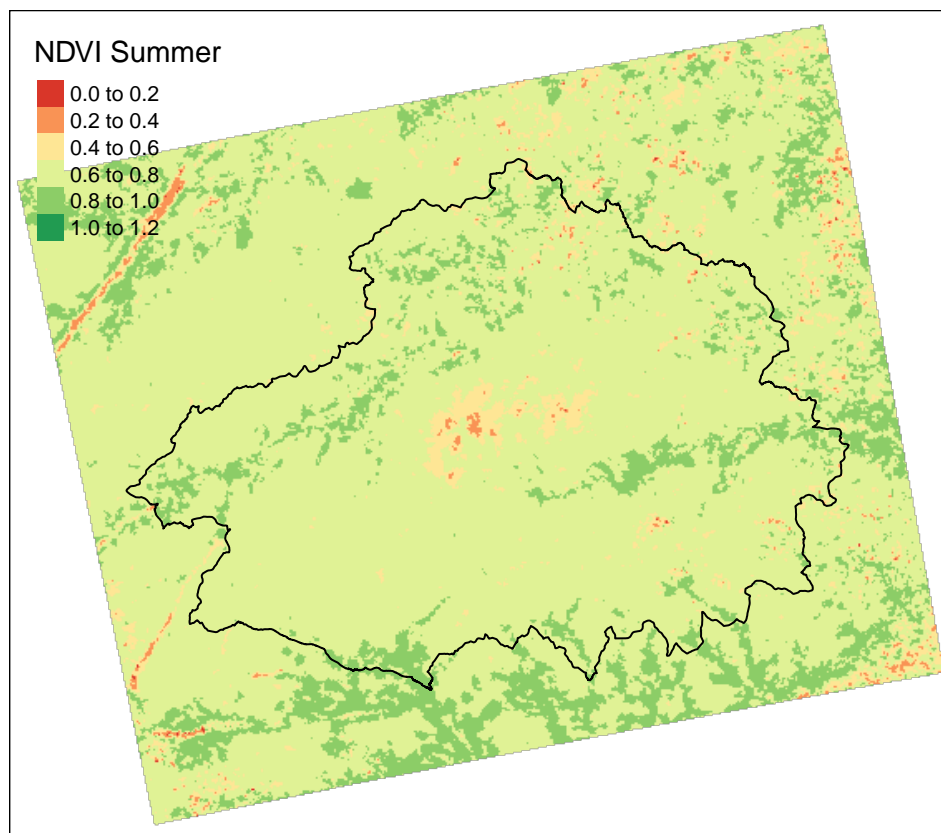
## tmap mode set to plotting

tm_shape(NDVI_1) +
  tm_raster(palette = "RdYlGn",
            title = "NDVI Winter", midpoint = NA) +
```

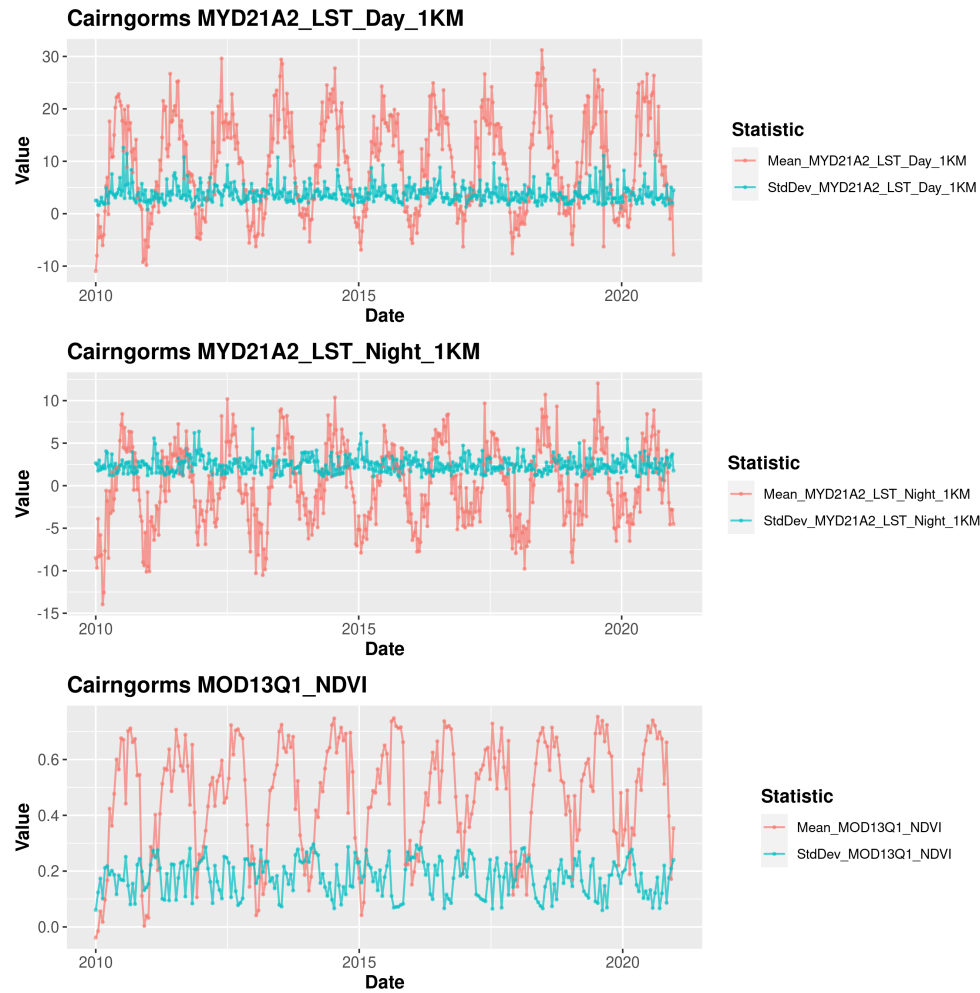
```
tm_shape(site_sf) +  
tm_borders("black", lwd = 1.0)
```



```
tm_shape(NDVI_2) +  
tm_raster(palette = "RdYlGn",  
          title = "NDVI Summer", midpoint = NA) +  
tm_shape(site_sf) +  
tm_borders("black", lwd = 1.0)
```



```
site_timeseries = list.files(file.path(Figures_dir, site),  
                             pattern = ".png$",  
                             full.names = TRUE)  
knitr::include_graphics(site_timeseries)
```



Busetto, Lorenzo, and Luigi Ranghetti. 2016. "MODISstp: An R Package for Preprocessing of Modis Land Products Time Series." *Computers & Geosciences* 97: 40–48. <https://doi.org/10.1016/j.cageo.2016.08.020>.