

# MODIS<sub>tsp</sub>: A tool for automatic preprocessing of MODIS time series - v1.3.0

*Lorenzo Busetto (busetto.l@irea.cnr.it), Luigi Ranghetti (ranghetti.l@irea.cnr.it)*

*2017-02-13*

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Installation</b>	<b>2</b>
<b>3</b>	<b>Running the tool in Interactive Mode: the MODIS<sub>tsp</sub> GUI</b>	<b>3</b>
<b>4</b>	<b>Running the tool in non-interactive mode and scheduled processing</b>	<b>8</b>
<b>5</b>	<b>Output formats and naming conventions</b>	<b>10</b>
<b>6</b>	<b>Accessing and analyzing the processed time series from R</b>	<b>10</b>
<b>7</b>	<b>Citation</b>	<b>11</b>
<b>8</b>	<b>Installing R and GDAL</b>	<b>11</b>
	<b>References</b>	<b>12</b>

## 1 Introduction

MODIS<sub>tsp</sub> is a novel “R” package allowing to automatize the creation of time series of rasters derived from MODIS Land Products data. It allows to perform several preprocessing steps on MODIS data available within a given time period.

Development of MODIS<sub>tsp</sub> started from modifications of the ModisDownload “R” script by Thomas Hengl (2010), and successive adaptations by Babak Naimi (2014). The basic functionalities for download and preprocessing of MODIS datasets provided by these scripts were gradually incremented with the aim of:

- developing a standalone application allowing to perform several preprocessing steps (e.g., download, mosaicking, reprojection and resize) on all available MODIS land products by exploiting a powerful and user-friendly GUI front-end;
- allowing the creation of time series of both MODIS original layers and additional Quality Indicators (e.g., data acquisition quality, cloud/snow presence, algorithm used for data production, etc. ) extracted from the aggregated bit-field QA layers;
- allowing the automatic calculation and creation of time series of several additional Spectral Indexes starting from MODIS surface reflectance products.

All processing parameters can be easily set with a user-friendly GUI, although non-interactive execution exploiting a previously created Options File is possible. Stand-alone execution outside an “R” environment is also possible, allowing to use scheduled execution of MODIS<sub>tsp</sub> to automatically update time series related to a MODIS product and extent whenever a new image is available.

Required MODIS HDF files are automatically downloaded from NASA servers and resized, reprojected, resampled and processed according to user’s choices. For each desired output layer, outputs are saved as

single-band rasters corresponding to each acquisition date available for the selected MODIS product within the specified time period. “R” *RasterStack* objects with temporal information as well as Virtual raster files (GDAL vrt and ENVI META files) facilitating access to the entire time series can be also created.

## 2 Installation

**IMPORTANT:** *MODISTsp* requires R  $v \geq 3.2.1$  and GDAL (Geospatial Data Abstraction Library)  $v \geq 1.11.1$  To be installed in your system. Brief instructions for installing R and GDAL can be found [HERE](#).

### 2.1 On Windows

1. Install and load the `gWidgetsRgtk2` package:

```
install.packages("gWidgetsRgtk2")  
library(gWidgetsRgtk2)
```

Upon loading the package, an error window will probably appear. **Don’t worry!** This is just signaling that *libatk-1.0-0.dll* is missing from your system. This is due to the fact that library “GTK+” is not yet installed on your system and needs to be installed. To do so, press “OK”. A new window dialog window will appear, asking if you want to install “GTK+”. Select “Install GTK+” and then “OK”. Windows will download and install the GTK+ library. When it finishes, the RSession will be restarted and you should be ready to go!<sup>1</sup>

2. Install *MODISTsp* package from GitHub (you’ll need to have the “devtools” package installed and loaded).

```
install.packages("devtools")  
library(devtools)  
install_github("lbusett/MODISTsp")
```

### 2.2 On Linux systems

1. Install the following required dependencies:

- Cairo  $\geq 1.0.0$ , ATK  $\geq 1.10.0$ , Pango  $\geq 1.10.0$ , GTK+  $\geq 2.8.0$ , GLib  $\geq 2.8.0$  (required by package `RGtk2`)
- Curl (required by package `curl`)
- GDAL  $\geq 1.6.3$ , PROJ.4  $\geq 4.4.9$  (required by package `rgdal`)

On *Debian and Ubuntu-based* systems, to install packages open a terminal and type

```
sudo apt-get install r-cran-cairodevice r-cran-rgtk2 libcairo2-dev libatk1.0-dev libpango1.0-dev lib
```

On *rpm-base systems*, to install packages open a terminal and type

```
sudo yum install libcairo2-devel libatk1.0-devel libpango1.0-devel gtk2-devel  
glib2-devel libcurl4-devel gdal-devel proj-devel
```

2. From R install the libraries `gWidgetsRgtk2` and `devtools`:

```
install.packages(c("devtools", "gWidgetsRgtk2"))
```

3. Install *MODISTsp* package from GitHub (you’ll need to have the “devtools” package loaded):

---

<sup>1</sup>If you encounter problems installing the `gWidgetsRgtk2` library, please signal it in the issues GitHub page of *MODISTsp* and we’ll try to help you!

```
library(devtools)
install_github("lbusett/MODISrsp")
```

## 2.3 Dependencies

MODISrsp exploits functionalities of several other “R” packages. In particular, the following packages are imported:

`bitops` ( $\geq 1.9.6$ ), `data.table` ( $\geq 1.9.6$ ), `gdalUtils` ( $\geq 2.0.1.7$ ), `gWidgets` ( $\geq 1.0.5$ ), `hash` ( $\geq 2.2.6$ ), `plyr` ( $\geq 1.8.3$ ), `raster` ( $\geq 2.5-2$ ), `RCurl` ( $\geq 1.95-4.8$ ), `rgdal` ( $\geq 1.1-8$ ), `rgeos` ( $\geq 0.3-8$ ), `xts` ( $\geq 1.0-10$ ), `sp` ( $\geq 1.2-2$ ), `stringr` ( $\geq 1.0.0$ ), `XML` ( $\geq 3.98-1.1$ ), `xml2` ( $\geq 0.1.2$ ), `httr` ( $\geq 1.1.0$ ), `RJSONIO` ( $\geq 1.3.0$ )

while the following are suggested:

`knitr`, `rmarkdown`, `png`, `grid`, `gWidgetsRgtk2` ( $\geq 1.0-7$ ).

Additionally, MODISrsp requires availability of the GDAL - Geospatial Data Abstraction Library v. >1.11.1, with support for HDF4 raster format.

## 3 Running the tool in Interactive Mode: the MODISrsp GUI

To run the tool in interactive mode, load the package and launch the MODISrsp function, with no parameters:

```
library(MODISrsp)
MODISrsp()
```

This opens a GUI from which processing options can be specified and eventually saved (or loaded).<sup>2</sup> A description of the different processing parameters to be selected is reported in the “Selection of Processing parameters” subsection.

*Note:* PCs with a small screen can not be able to visualise the whole GUI; in this case, the user can add scroll bars with `MODISrsp(scrollWindow=TRUE)`.

### 3.1 Selection of processing parameters

When running MODISrsp in Interactive Mode, a user-friendly GUI allows selection of all processing options required for the creation of the desired MODIS time series. The main available processing options are described in detail in the following.

### 3.2 MODIS Product, Platform and Layers

Allows to select the MODIS product of interest using the “**Category**” and “**Product**” drop down menus. The user can then select which **MODIS platform(s)** should be considered for download and creation of the time series (“Terra”, “Aqua” or “Both”), and which **version** of the product - 5 or 6 (when available) - should be considered. After selecting the product, pushing the “**Click to Select Desired Layers**” button opens the “Select Processing Layers” GUI panel, from which the user **must** select which MODIS original and/or derived QI and SI layers should be processed:

---

<sup>2</sup>At the first execution of *MODISrsp*, a Welcome screen will appear, signaling that MODISrsp is searching for a valid GDAL installation. Press “ok” and wait for GDAL to be found. If nothing happens for a long time (e.g., several minutes), MODISrsp (and in particular the `gdalUtils` package on which it relies) is not finding a valid GDAL installation in the more common locations. To solve the problem: i) ensure that GDAL is properly installed in your system; ii) (on Windows) if it is installed, verify that GDAL is in your system PATH. and that the `GDAL_DATA` environment variable is correctly set (You can find simple instructions [HERE](#)); iii) if nothing works, signal it in the issues GitHub page of MODISrsp and we’ll try to help!

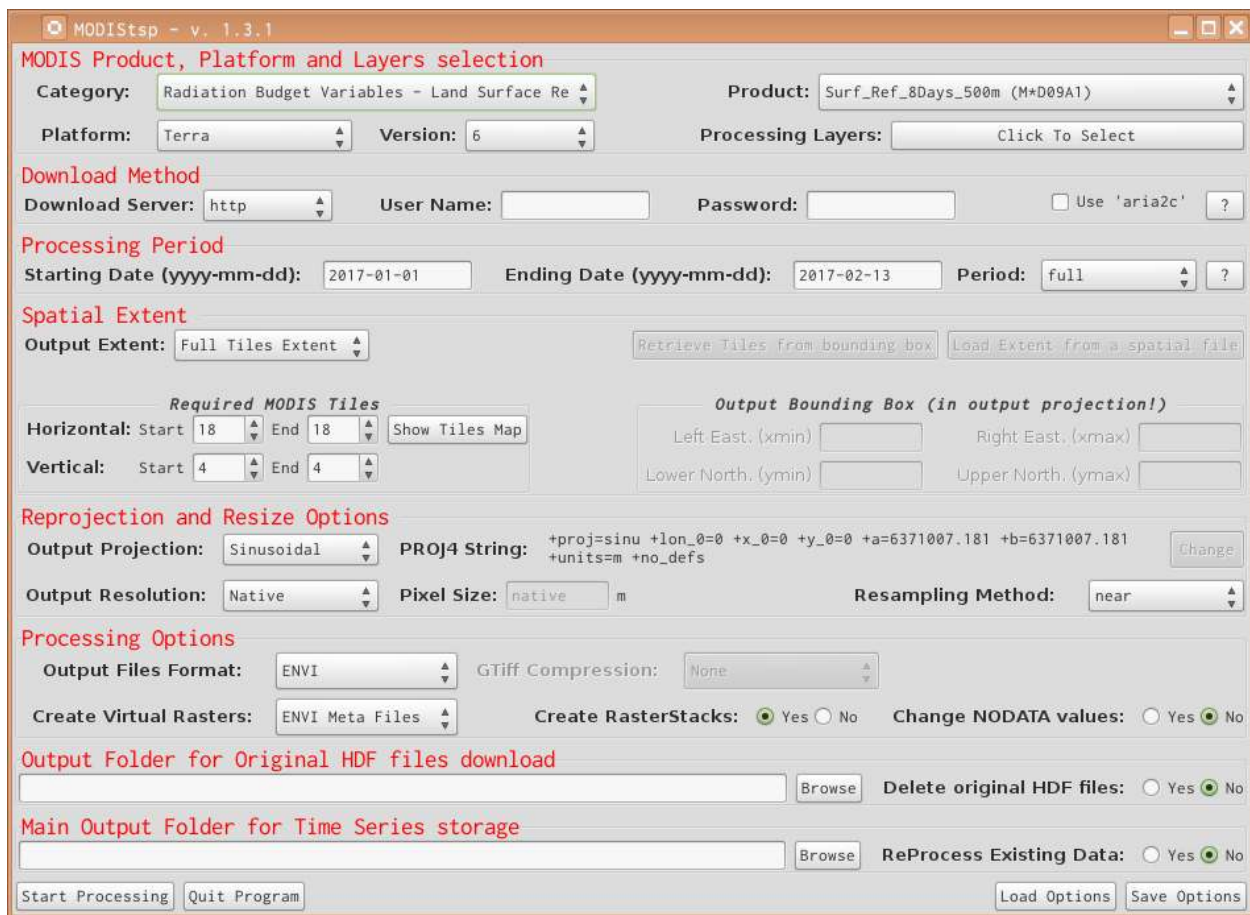


Figure 1: The MODISTsp main GUI

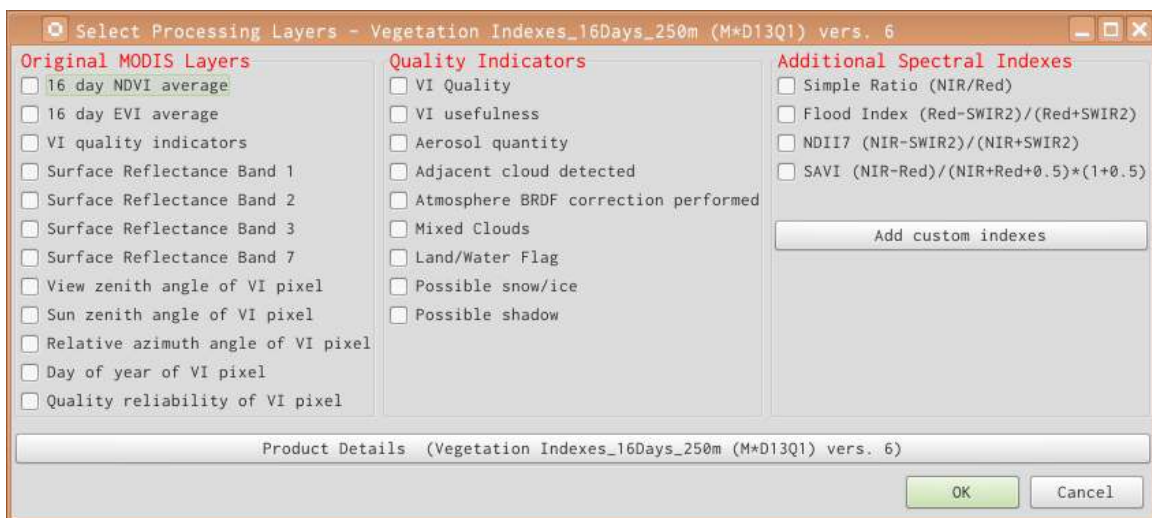


Figure 2: The "Select Processing Layers" GUI

1. The left-hand frame allows to select which original MODIS layers should be processed.
2. The central frame allows to select which Quality Indicators should be extracted from the original MODIS Quality Assurance layers.
3. For MODIS products containing surface reflectance data, the right-hand frame allows to select which additional Spectral Indexes should be computed.<sup>3</sup>

Some of the most commonly used Spectral Indexes are available for computation by default (Table II).

**Table II: List of default Spectral Indexes available in MODISTsp**

Index Acronym	Index name and reference
NDVI	Normalized Difference Vegetation Index (Rouse et al. 1973)
EVI	Enhanced Vegetation Index (A. Huete et al. 2002)
SR	Simple Ratio(Tucker 1979)
NDFI	Normalized Difference Flood Index (Boschetti et al. 2014)
NDII7 (NDWI)	Normalized Difference Infrared Index - Band 7 (Hunt and Rock 1989)
SAVI	Soil Adjusted Vegetation Index (A.R Huete 1988)
NDSI	Normalized Difference Snow Index (Hall et al. 2002)
NDII6	Normalized Difference Infrared Index - band 6 (Hunt and Rock 1989)
GNDVI	Green Normalized Difference Vegetation Index (Gitelson and Merzlyak 1998)
RGRI	Red Green Ratio Index (Gamon and Surfus 1999)
GRVI	Green-Red ratio Vegetation Index (Tucker 1979)

<sup>3</sup>The lists of original MODIS layers, QIs and Sis available for the selected product are automatically retrieved from the “MODISTsp\_Products\_Opts” XML file distributed with the package in /ExtData subfolder.

Users can however **specify other SIs to be computed without modifying MODISTsp source code** by clicking on the “**Add Custom Index**” button, which allow to provide info related to the new desired SI using a simple GUI interface.

Figure 3: The GUI for insertion of additional Spectral Indexes

Provided information (e.g., correct bandnames, computable formula, etc...) is automatically checked upon clicking “Set New Index”. On success, the new index is added in the list of available ones for all products allowing its computation. Clicking “Done !” returns to the main.

### 3.3 Download Method

Allows to specify which download method should be used. Available choices are:

1. **http**: download through ftp from NASA lpdaac http archive (<http://e4ftl01.cr.usgs.gov>). This requires providing a user name and password, which can be obtained by registering an account at the address <https://urs.earthdata.nasa.gov/profile>;
2. **ftp**: download from NASA ftp archive (<ftp://ladsweb.nascom.nasa.gov/>);
3. **offline**: this option allows to process/reprocess HDF files already available on the user’s PC without downloading from NASA – useful if the user already has an archive of HDF images, or to reprocess data already downloaded via MODISTsp (for example, to create time series for an additional layer).

The **use\_aria2c** option allows to accelerate the download from NASA archives, provided that the “aria2c” software is installed in your system. To download it and install it, see: <https://aria2.github.io/>

### 3.4 Processing Period

Allows to specify the starting and ending dates to be considered for the creation of the time series. Dates **must** be provided in the *yyyy-mm-dd* format (e.g., 2015-01-31)

The **Period** dropdown menu allows to choose between two options:

1. **full**: all available images between the starting and ending dates are downloaded and processed;
2. **seasonal**: allows to download data for only one part of the year, but for multiple years (for example, if the starting date is 2005-03-01 and the ending is 2010-06-01, only the images of March, April and May for the year between 2005 and 2010 will be downloaded). This allows to easily process data concerning a particular season of interest.

### 3.5 Spatial Extent

Allows to define the area of interest for the processing. Two main options are possible:

1. *Full Tiles Extent*: the user must specify which MODIS tiles he would like to process using the “Start” and “End” horizontal and vertical sliders in the “Required MODIS Tiles” frame. During processing, data from the different tiles is mosaiced, and a single file covering the total area is produced for each acquisition date (Note: pressing the “show map” button, a representation of the MODIS tiles grid is shown to facilitate the selection).
2. *Resized*: the user can specify the spatial extent of the desired outputs either by:
  - a. **manually inserting** the coordinates of the Upper Left and Lower Right corners of the area of interest in the “Bounding Box” frame. *Coordinates of the corners must be provided in the coordinate system of the selected output projection*;
  - b. **pressing the “Load Extent from a Spatial File” and selecting a raster or vector spatial file**. In this case, the bounding box of the selected file is retrieved, converted in the selected output projection, and shown in the “Bounding Box” frame. Required input MODIS tiles are also automatically retrieved from the output extent, and the tiles selection sliders modified accordingly.

### 3.6 Reprojection and Resize

Allows to specify the options to be used for reprojecting and resizing the MODIS images. In particular:

1. the “**Output Projection**” menu allows to either select one of the pre-defined output projections or specify a user-defined one by selecting “User Defined” and then inserting a valid “Proj4” string in the pop-up window. Validity of the Proj4 string is automatically checked, and error messages issued if the check fails;
2. the “**Output Resolution**”, “**Pixel Size**” and “**Reprojection Method**” menus allow to specify whether output images should inherit their spatial resolution from the original MODIS files, or be resampled to a user-defined resolution. In the latter case, output spatial resolution must be specified in the measure units of the selected output projection. At the moment, the resampling method can instead be chosen among “Nearest Neighbour” and “Mode” (Useful for downsampling purposes). Other resampling methods (e.g., bilinear, cubic) are not currently supported since i) they cannot be used for resampling of categorical variables such as the QA and QI layers, and ii) using them on continuous variable (e.g., reflectance, VI values) without performing an a-priori data cleaning would risk to contaminate the values of high-quality observations with those of low-quality ones.

### 3.7 Processing Options

Allows first of all to specify the **format desired for the output images**. Two of the most commonly formats used in remote sensing applications are available at the moment: ENVI binary and GeoTiff. If GeoTiff is selected, the type of file compression can be also specified among “None”, “PACKBITS”, “LZW” and “DEFLATE”.

The user can then specify if **virtual multitemporal files** should be created. These virtual files allow access to the entire time series of images as a single file without the need of creating large multitemporal raster images. Available virtual files formats are ENVI metafiles and GDAL “vrt” files. Additionally, the user can select if he desires to save the time series also as “R” *rasterStack* objects (with temporal information added through the “setZ” method of the *raster* package). This may be useful in order to easily access the preprocessed MODIS data within “R” scripts.

Finally, users can select if the **NoData values** of MODIS layers should be kept at their original values, or changed to those specified within the “MODIS<sub>tp</sub>\_Products\_Opts” XML file. By selecting “Yes” in the “Change Original NODATA values” checkbox, NoData of outputs are set to the largest integer value possible for the data type of the processed layer (e.g., for 8-bit unsigned integer layers, NoData is set always to 255, for 16-bit signed integer layers to 32767, and for 16-bit unsigned integer layers to 65535). Information about the new NoData values is stored both in the output rasters, and in the XML files associated with them.

### 3.8 Main Output Folder for Time Series Storage

Allows to specify the main folder where the pre-processed time series data will be saved stored. The **“Reprocess Existing Data”** checkbox allows to specify if images already available should be reprocessed if a new run of MODISTsp is launched with the same output folder. If set to “No”, MODISTsp skips dates for which output files following the MODISTsp naming conventions are already present in the output folder. This allows to incrementally extend MODIS time series without reprocessing already available dates.

### 3.9 Output Folder for Original HDF Storage

Allows to specify the folder where downloaded **original MODIS HDF files** are stored. The **“delete original HDF files”** checkbox allows to specify if the downloaded images should be deleted from the file system at the end of the processing. To avoid accidental file deletion, this is always set to “No” by default, and a warning is issued before execution whenever the selection is changed to “Yes”.

## 4 Running the tool in non-interactive mode and scheduled processing

MODISTsp can be also launched in non-interactive mode by setting the optional “GUI” parameter to FALSE, and the “options\_file” parameter to the path of a previously saved Options file. This allows to exploit MODISTsp functionalities within generic “R” processing scripts

```
library(MODISTsp)
# --> Specify the path to a valid options file saved in advance
options_file = "yourpath/youroptions.json" # generated from the GUI
MODISTsp(gui = FALSE, options_File = options_file)
```

Single parameters can be also be directly passed to the MODISTsp() function; for example:

```
library(MODISTsp)
options_file = "yourpath/youroptions.json"
MODISTsp(gui = FALSE, options_File = options_file, download_server = "ftp")
```

will launch MODISTsp with the parameters saved in “yourpath/youroptions.json” options file, with the exception of “download\_server” parameter, which is set to “ftp”. Details about the parameter names can be found with ?MODISTsp.

Specifying also the “*spatial\_file\_path*” parameter overrides the output extent of the selected Options File. This allows to perform the same preprocessing on different extents using a single Options File, by looping on an array of spatial files representing the desired output extents.

For example:

```
# Create a character array containing a list of shapefiles (or other spatial files)
extent_list = list.files("path/containing/some/shapefiles/", "\\shp$")

# loop on the list of spatial files and run MODISTsp using each of them to automatically
# define the output extent (A separate output folder is created for each input file).

for (single_shape in extent_list) {
  MODISTsp(gui = FALSE, options_File = "yourpath/youroptions.json",
    spatial_file_path = single_shape )
}
```



## 4.1 Standalone execution and scheduled processing

MODISTsp can be executed as a standalone application using the MODISTsp.bat (for Windows) or MODISTsp.sh (for Linux) batch execution scripts available in the “MODISTsp/ExtData/launcher” subfolder of the package installation. Double-clicking the files or launching them from a shell without parameters launches MODISTsp in interactive mode.

Non-interactive mode is triggered by adding the “-g” argument to the call, and specifying the path to a valid Options File as “-s” argument

- **In Linux:**

```
path_to_MODISTsp_sh/MODISTsp.sh -g -s "/yourpath/youroptions.json"
```

(launch path\_to\_MODISTsp\_sh/MODISTsp.sh -h for details).

- **In Windows:**

```
path_to_MODISTsp.bat\MODISTsp.bat -g -s "X:/yourpath/youroptions.json"
```

(launch path\_to\_MODISTsp.bat\MODISTsp.bat -h for details).

Standalone non-interactive execution easily allows to automatically update the time series of a selected product over a given study area whenever a new MODIS image is available. To do that, the user must simply:

1. Open the MODISTsp GUI, define the parameters of the processing specifying a date in the future as the “Ending Date” and save the processing options. Then quit the program
2. Schedule non-interactive execution of MODISTsp.bat (or MODISTsp.sh) as windows scheduled task (or linux “cron” job) according to a specified time schedule, specifying the path of a previously saved Options file as additional argument:

- **In Linux:** edit your crontab by opening a terminal and type:

```
crontab -e
```

Then add an entry for the MODISTsp.bsh For example, if you want to run the tool every day at 23.00, add the following row:

```
0 23 * * * /bin/bash /path_to_MODISTsp_sh/MODISTsp.sh -g -s "/yourpath/youroptions.json"
```

- **In Windows:** create a Task following these instructions; add the path of the MODISTsp.bat launcher as Action (point 6), and specify -g -s "X:/yourpath/youroptions.json" as argument.

## 4.2 Adding links to desktop/Start menu for standalone execution

Links to the MODISTsp.bat or MODISTsp.sh standalone launchers can be created automatically launching from R the function MODISTsp\_install\_launcher()

- **In Linux:** this creates a desktop entry (accessible from the menu in the sections “Science” and “Geography”), and a symbolic link in a known path (default: /usr/bin/MODISTsp). If the path of the symbolic link is included in the user PATH variable, the standalone execution can be done simply calling MODISTsp -g -s "/yourpath/youroptions.json".
- **In Windows:** this creates a link in the Start Menu and optionally a desktop shortcut.

See ?install\_MODISTsp\_launcher for details and path customisations.

## 5 Output formats and naming conventions

Output raster files are saved in specific subfolders of the main output folder. A separate subfolder is created for each processed original MODIS layer, Quality Indicator or Spectral Index. Each subfolder contains one image for each processed date, created according to the following naming conventions:

“ProdCode”\_“Layer”\_“YYYY”\_“DOY”：“ext” (e.g., *MOD13Q1\_NDVI\_2000\_065.dat*)

**ProdCode** is the code name of the MODIS product from which the image was derived (e.g., MOD13Q1), **Layer** is a short name describing the dataset (e.g., b1\_Red, NDII, UI), **YYYY** and **DOY** corresponds to the year and DOY (Day of the Year) of acquisition of the original MODIS image, and ext is the file extension (.tif for GTiff outputs, or .dat for ENVI outputs).

ENVI and/or GDAL virtual time series files and *RasterStack* RData objects are instead stored in the “Time\_Series” subfolder if required.

Naming convention for these files is as follow:

“ProdCode”\_“Layer”\_“StartDOY”\_“StartYear”\_“EndDOY”\_“EndYear2”\_“suffix”.ext”

(*MOD13Q1\_NDVI\_49\_2000\_17\_2015\_RData.dat*)

where *suffix* indicates the type of virtual file (ENVI, GDAL or RData), while *StartDOY*, *StartYear*, *EndDOY* and *EndYear* indicate the temporal extent of the time serie created.

## 6 Accessing and analyzing the processed time series from R

Preprocessed MODIS data can be retrieved within R scripts either by accessing the single-date raster files, or by loading the saved *RasterStack* objects. This second option allows accessing the complete data stack and analyzing it using the functionalities for raster/raster time series analysis, extraction and plotting provided for example by the **raster** or **rasterVis** packages. *MODISTsp* provides however an efficient function *MODISTsp\_extract* for extracting time series data at specific locations. The function takes as input a *RasterStack* object with temporal information created by *MODISTsp*, the starting and ending dates for the extraction and a standard *\_Sp\** object (or an ESRI shapefile name) specifying the locations (points, lines or polygons) of interest, and provides as output a *\_xts* object containing time series for those locations. If the input is of class *SpatialPoints*, the output object contains one column for each point specified, and one row for each date. If it is of class *SpatialPolygons* (or *SpatialLines*), it contains one column for each polygon (or each line), with values obtained applying the function specified as the “FUN” argument (e.g., mean, standard deviation, etc.) on pixels belonging to the polygon (or touched by the line), and one row for each date.

As an example the following code:

```
#Set the input paths to raster and shape file
infile = 'in_path/MOD13Q1_MYD13Q1_NDVI_49_2000_353_2015_RData.RData'
shpname = 'path_to_file/rois.shp'
#Set the start/end dates for extraction
startdate = as.Date("2010-01-01")
enddate = as.Date("2014-12-31")
#Load the RasterStack
inrts = get(load(infile))
# Compute average and St.dev
dataavg = MODISTsp_extract(inrts, shpname, startdate, enddate, FUN = 'mean', na.rm = T)
datasd = MODISTsp_extract(inrts, shpname, startdate, enddate, FUN = 'sd', na.rm = T)
# Plot average time series for the polygons
plot.xts(dataavg)
```

loads a *RasterStack* object containing 8-days 250 m resolution time series for the 2000-2015 period and extracts time series of average and standard deviation values over the different polygons of a user's selected shapefile on the 2010-2014 period.

## 7 Citation

To cite MODISTsp please use:

L. Busetto, L. Ranghetti (2016) MODISTsp: An R package for automatic preprocessing of MODIS Land Products time series, *Computers & Geosciences*, Volume 97, Pages 40-48, ISSN 0098-3004, <http://dx.doi.org/10.1016/j.cageo.2016.08.020>, URL: <https://github.com/lbusett/MODISTsp>.

## 8 Installing R and GDAL

### 8.1 Installing R

#### 8.1.1 Windows

Download and install the latest version of R which can be found [here](#).

#### 8.1.2 Linux

Please refer to the documentation which can be found [here](#), opening the directory relative to the user Linux distribution. The documentation provides instruction to add CRAN repositories and to install the latest R version. With Ubuntu 15.10 Wily (and newer) this step is not mandatory (although recommended), since packaged version of R is  $\geq 3.2.1$  (although not the latest); in this case, user can install R by simply typing in a terminal

```
sudo apt-get install r-base
```

### 8.2 Installing GDAL $\geq 1.11.1$

#### 8.2.1 Windows

The easiest way to install GDAL on Windows is from the OSGeo4W Website

1. Open the OSGeo4W Website
2. In the **Quick Start for OSGeo4W Users** section, select the download of 32bit or 64bit of OSGeo4W network installer
3. Run the installer
  - *Easiest Option:*
    - Select **Express Desktop Install**, then proceed with the installation. This will install GDAL and also other useful Spatial Processing softwares like QGIS and GRASS GIS
  - *Advanced Option:*
    - Select **Advanced Install**, then click on “Next” a few times until you reach the “Select Packages” screen.
    - Click on “Commandline Utilities\_”, and on the list look for “\_gdal: The GDAL/OGR library...” entry
    - Click on “Skip”: the word “skip” will be replaced by the current GDAL version number
    - Click on “Next” a few times to install GDAL

## 8.2.2 Debian and Ubuntu-based systems

1. Ensure that your repositories contain a version of `gdal-bin`  $\geq 1.11.1$ . In particular, official repositories of Ubuntu 15.04 Vivid (or older) and Debian Jessie (or older) provide older versions of GDAL, so it is necessary to add UbuntuGIS-unstable repository before installing. To do this, follow instructions here). With Ubuntu 15.10 Wily (and newer) this step is not mandatory, although recommended in order to have updated version of GDAL installed.
2. To install GDAL a terminal and type

```
sudo apt-get install gdal-bin
```

## 8.2.3 ArchLinux

GDAL is maintained updated to the latest version as binary package within the community repository; although that, the support for HDF4 format is not included. To bypass this problem, ArchLinux users can install `gdal-hdf4` package from AUR (see here or here for the package installation from AUR). This package is updated manually after each release of `gdal` on the community repository, so a temporal shift between a new `gdal` release and the update of `gdal-hdf4` could happen. If you want to manually add the support for HDF4 in case `gdal-hdf4` is out-of-date, you can do it following these instructions.

## 8.2.4 Other Linux systems

Install the packaged binary of GDAL included in your specific distribution; if the version is older than 1.11.1, or if the support for HDF4 format is not included, you can manually install the HDF4 library and compile the source code by adding the parameter `--with-hdf4` to the `configure` instruction).

## References

- Boschetti, M., F. Nutini, G. Manfron, P. A. Brivio, and A. Nelson. 2014. "Comparative analysis of normalised difference spectral indices derived from MODIS for detecting surface water in flooded rice cropping systems." *PLoS ONE* 9 (2). doi:10.1371/journal.pone.0088741.
- Gamon, J.A., and J.S. Surfus. 1999. "Assessing leaf pigment content and activity with a reflectometer." *New Phytologist* 143 (1): 105–17.
- Gitelson, A.A., and M.N. Merzlyak. 1998. "Remote sensing of chlorophyll concentration in higher plant leaves." *Advances in Space Research* 22 (5): 689–92. doi:10.1016/S0273-1177(97)01133-2.
- Hall, Dorothy K, George A Riggs, Vincent V Salomonson, Nicolo E DiGirolamo, and Klaus J Bayr. 2002. "MODIS snow-cover products." *Remote Sensing of Environment* 83 (1-2): 181–94. doi:10.1016/S0034-4257(02)00095-0.
- Hengl, T. 2010. "Download and resampling of MODIS images." [http://spatial-analyst.net/wiki/index.php?title=Download\\_and\\_resampling\\_of\\_MODIS\\_images](http://spatial-analyst.net/wiki/index.php?title=Download_and_resampling_of_MODIS_images).
- Huete, A., K. Didan, T. Miura, E.P. Rodriguez, X. Gao, and L.G. Ferreira. 2002. "Overview of the radiometric and biophysical performance of the MODIS vegetation indices." *Remote Sensing of Environment* 83 (1-2): 195–213. doi:10.1016/S0034-4257(02)00096-2.
- Huete, A.R. 1988. "A soil-adjusted vegetation index (SAVI)." *Remote Sensing of Environment* 25 (3): 295–309. doi:10.1016/0034-4257(88)90106-X.
- Hunt, J.R., and B. Rock. 1989. "Detection of changes in leaf water content using Near- and Middle-Infrared

reflectances.” *Remote Sensing of Environment* 30 (1): 43–54. doi:10.1016/0034-4257(89)90046-1.

Naimi, B. 2014. “ModisDownload: an R function to download, mosaic, and reproject the MODIS images.” <http://r-gis.net/?q=ModisDownload>.

Rouse, J.W.J., R.H. Haas, J.A. Schell, and D.W. Deering. 1973. “Monitoring vegetation systems in the Great Plains with ERTS. Third ERTS Symposium, NASA SP-351. U.S. Gov. Printing office.” Edited by S.C. Freden, E.P. Mercanti, and M.A. Becker. NASA.

Tucker, C.J. 1979. “Red and photographic infrared linear combinations for monitoring vegetation.” *Remote Sensing of Environment* 8 (2): 127–50. doi:10.1016/0034-4257(79)90013-0.