

# Package ‘MODISp’

August 17, 2017

**Title** A Tool for Automating Download and Preprocessing of MODIS Land Products Data

**Type** Package

**Version** 1.3.3.1

**Description** Allows automating the creation of time series of rasters derived from MODIS Satellite Land Products data. It performs several typical preprocessing steps such as download, mosaicking, reprojection and resize of data acquired on a specified time period. All processing parameters can be set using a user-friendly GUI. Users can select which layers of the original MODIS HDF files they want to process, which additional Quality Indicators should be extracted from aggregated MODIS Quality Assurance layers and, in the case of Surface Reflectance products, which Spectral Indexes should be computed from the original reflectance bands. For each output layer, outputs are saved as single-band raster files corresponding to each available acquisition date. Virtual files allowing access to the entire time series as a single file are also created. Command-line execution exploiting a previously saved processing options file is also possible, allowing to automatically update time series related to a MODIS product whenever a new image is available.

**License** GPL-3

**Depends** R (>= 3.1.3)

**Imports** bitops (>= 1.0-6), data.table (>= 1.9.6), gdalUtils (>= 2.0.1.7), gWidgets (>= 0.0-54), gWidgetsRGtk2 (>= 0.0-83), hash (>= 2.2.6), httr (>= 1.1.0), pacman, plyr (>= 1.8.3), raster (>= 2.5-2), RCurl (>= 1.95-4.8), rgdal (>= 1.0-3), rgeos (>= 0.3-8), RJSONIO (>= 1.3.0), sp (>= 1.2-2), stringr (>= 1.0.0), xts (>= 0.9-7), XML (>= 3.98-1.1), parallel

**Suggests** knitr, rmarkdown, png, grid, testthat, covr

**SystemRequirements** Cairo >= 1.0.0, ATK (>= 1.10.0), Pango (>= 1.10.0), GTK+ (>= 2.8.0), GLib (>= 2.8.0), Curl, GDAL (>= 1.6.3), PROJ.4 (>= 4.4.9)

**URL** <https://github.com/lbusett/MODISp>

**BugReports** <https://github.com/lbusett/MODISp/issues>

**LazyData** true  
**VignetteBuilder** knitr  
**RoxygenNote** 6.0.1  
**NeedsCompilation** no  
**Author** Lorenzo Busetto [aut, cre],  
 Luigi Ranghetti [aut]  
**Maintainer** Lorenzo Busetto <lbusett@gmail.com>  
**Repository** CRAN  
**Date/Publication** 2017-08-17 09:04:17 UTC

## R topics documented:

bbox_from_file . . . . .	2
install_MODISStsp_launcher . . . . .	3
lpdaac_getmod_dates . . . . .	4
lpdaac_getmod_dirs . . . . .	5
lpdaac_getmod_names . . . . .	6
MODISStest_check_md5 . . . . .	7
MODISStsp . . . . .	8
MODISStsp_addindex . . . . .	9
MODISStsp_check_files . . . . .	11
MODISStsp_extract . . . . .	13
MODISStsp_GUI . . . . .	15
MODISStsp_process . . . . .	16
MODISStsp_process_indexes . . . . .	19
MODISStsp_process_QA_bits . . . . .	20
MODISStsp_read_xml . . . . .	21
MODISStsp_vrt_create . . . . .	22
reproj_bbox . . . . .	23
<b>Index</b>	<b>24</b>

---

bbox_from_file	<i>bbox_from_file</i>
----------------	-----------------------

---

## Description

Helper function used to retrieve the bounding box of a specified spatial file recognized by GDAL or OGG: the function reads the extent using gdalinfo or ogrinfo, converting it to a specified CRS.

## Usage

```
bbox_from_file(file_path, out_crs)
```

**Arguments**

file_path	The path of the spatial file.
out_crs	The output projection (string format).

**Note**

License: GPL 3.0

**Author(s)**

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

install\_MODISrsp\_launcher  
*install\_MODISrsp\_launcher*

---

**Description**

Function which allows to use MODISrsp in batch mode by creating links

**Usage**

```
install_MODISrsp_launcher(bin_dir = NA, desktop_dir = NA,
  desktop_shortcut = TRUE, sudo = FALSE)
```

**Arguments**

bin_dir	in Linux, directory in which the link to the bash script should be placed (default: /usr/bin; a path included in the PATH environment variable is suggested); in Windows, directory where to place the menu entry in the Start Menu (default: Start Menu -> Programs -> MODISrsp).
desktop_dir	if desktop_shortcut=TRUE: in Linux, directory in which the desktop entry should be placed (default: /usr/share/applications); in Windows, directory where to place the desktop entry (default: Desktop).
desktop_shortcut	logical value which indicates if the desktop entry or the desktop shortcut should be created (default: TRUE).
sudo	(Linux only) logical value which indicates if administrator rights have to be used to write within bin_dir and desktop_dir (default: FALSE); in this case, the root password is requested when launching the function. Note that default values of bin_dir and desktop_dir requires to set this option to TRUE (or to launch the script in a root session of R)!

**Details**

MODISStp tool can be used also as a standalone tool by launching a bash/batch script, which is stored in the function files. In order to simply retrieve it, this function will create a desktop entry and a symbolic link to the bash script (in Linux) or a link in the Start Menu to the batch script (in Windows). Note that, if the packages MODISStp is installed in a version-dependent directory (as the default one is), this function should be re-executed after an R upgrade, otherwise the links would continue to point to the old package version!

**Note**

License: GPL 3.0

**Author(s)**

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

**Examples**

```
# Linux: common installation (script in /usr/bin,
# desktop entry in /usr/share/applications)
# (requires administrator permissions)
## Not run:
install_MODISStp_launcher(sudo = TRUE)
# the administrator password is asked interactively
## End(Not run)

# Linux: installation in a directory which does
# not require administrator permissions
## Not run:
install_MODISStp_launcher(bin_dir = "~/bin"), desktop_dir = "~/Desktop"
## End(Not run)

# Windows: common installation
# (script in the Start Menu and shortcut on the desktop)
## Not run:
install_MODISStp_launcher()
## End(Not run)
```

---

lpdaac\_getmod\_dates    *lpdaac\_getmod\_dates*

---

**Description**

Accessory function to find the folders corresponding to the requested dates period within the full list retrieved by lpdaac\_getmod\_dirs

**Usage**

```
lpdaac_getmod_dates(dates, date_dirs)
```

**Arguments**

dates	2- element string array specifying start/end dates (yyyy.mm.dd) for which the http addresses of folders in lpdaac should be retrieved (e.g., c("2015.1.1", "2015.12.31"))
date_dirs	data frame full list of folders in lpdaac archive for product of interest

**Value**

array of folder names containing data for the MODIS product acquired in the period specified by "dates"

**Note**

License: GPL 3.0

**Author(s)**

Luigi Ranghetti, PhD (2016) <ranghetti.l@irea.cnr.it>

---

lpdaac\_getmod\_dirs      *lpdaac\_getmod\_dirs*

---

**Description**

Accessory function to get the full list of directories on the lpdaac http site (modified after Barry Rowlingson function):

**Usage**

```
lpdaac_getmod_dirs(ftp, http, used_server = NA, user = user,
  password = password, gui, out_folder_mod, .Platform)
```

**Arguments**

ftp	string ftp site corresponding to a given MODIS product
http	string http site on lpdaac corresponding to a given MODIS product
used_server	string can assume values "http" or "ftp" depending on the used download server; if NA, the script tries to download with http, using ftp if the download fails
user	username for earthdata server
password	password for earthdata server
gui	logical indicates if processing was called within the GUI environment or not. If not, direct processing messages to the log
out_folder_mod	output folder for original HDF storage
.Platform	string os platform (from call to .Platform)

**Value**

list of all available folders (a.k.a. dates) for the requested MODIS product on lpdaac archive

**Note**

License: GPL 3.0

**Author(s)**

Original code by Babak Naimi (`.getModisList`, in `ModisDownload.R`) modified to adapt it to MODISstsp scheme and to http archive (instead than old FTP) by:

Lorenzo Busetto, PhD (2014-2017) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2016-2017) <ranghetti.l@irea.cnr.it>

---

lpdaac\_getmod\_names    *lpdaac\_getmod\_names*

---

**Description**

Accessory function to find the names of HDF images corresponding to a given date and interval of spatial tiles within the lpdaac archive

**Usage**

```
lpdaac_getmod_names(http, ftp, used_server, user, password, date_dir, v, h,
  tiled, out_folder_mod, gui)
```

**Arguments**

http	string http site on lpdaac corresponding to a given MODIS product
ftp	string ftp site corresponding to a given MODIS product
used_server	string can assume values "http" or "ftp" depending on the used download server; it cannot be NA
user	username for earthdata server
password	password for earthdata server
date_dir	array of folder names containing data for the MODIS product acquired in a given period (return array from <code>lpdaac_getmod_dates</code> )
v	int. array containing a sequence of the vertical tiles of interest (e.g., <code>c(18,19)</code> )
h	int. array containing a sequence of the horizontal tiles of interest (e.g., <code>c(3,4)</code> )
tiled	0/1 1 = tiled product; 0 = non-tiled product (resolution 0.05 deg)
out_folder_mod	output folder for original HDF storage
gui	logical indicates if processing was called within the GUI environment or not. If not, direct processing messages to the log

**Value**

Modislist names of HDF images corresponding to the requested tiles available for the product in the selected date

**Note**

License: GPL 3.0

**Author(s)**

Original code by Babak Naimi (.getModisList, in [ModisDownload.R](#)) modified to adapt it to MODIS<sub>test</sub> scheme and to http archive (instead than old FTP) by:

Lorenzo Busetto, PhD (2014-2016) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2016) <ranghetti.l@irea.cnr.it>

---

MODIS<sub>test</sub>\_check\_md5    *MODIS<sub>test</sub>\_check\_md5*

---

**Description**

internal function which tests if the output of a test is corresponds to the expected one.

**Usage**

MODIS<sub>test</sub>\_check\_md5(test)

**Arguments**

test                    integer number of the performed test

**Details**

When running in test mode, MODIS<sub>test</sub> is expected to produce the same files

**Note**

License: GPL 3.0 license GPL 3.0

**Author(s)**

Luigi Ranghetti, PhD (2017) <ranghetti.l@irea.cnr.it>

---

 MODIS<sub>tsp</sub>


---

*MODIS<sub>tsp</sub>*


---

## Description

Main function for the MODIS Time Series Processing Tool (MODIS<sub>tsp</sub>)

## Usage

```
MODIStsp(gui = TRUE, options_file = NULL, spatial_file_path = NULL,
         scrollWindow = FALSE, test = -1)
```

## Arguments

<code>gui</code>	logical parameter (TRUE: the GUI is opened before processing; FALSE: the saved parameters are retrieved from "options_file")
<code>options_file</code>	settings (optional): full path of the JSON file containing the processing options (default: MODIS <sub>tsp</sub> _Previous.json in subfolder Previous);
<code>spatial_file_path</code>	(optional): full path of a spatial file to use as extent (default=NULL): if defined, the processing options which define the extent, the selected tiles and the "Full Tile / Resized" options are not considered; instead, new files are created on the extent of the provided spatial file.
<code>scrollWindow</code>	(optional) logical parameter: if TRUE, the GUI window is opened fullscreen with scrollbars (this is useful on devices with small display). If using a device with a display resolution $\geq 1024 \times 768$ , leaving this parameter to FALSE is suggested.
<code>test</code>	(optional) integer: if set, the tool is run in test mode, using a preset Option File instead than opening the GUI or accepting the option_file parameter. The number indicates which preset file to be used (five files are available). If test=0, the files are selected randomly. Default value (-1) indicates that the tool is executed normally (not in test mode). This modality is useful to test the tool in case of errors.

## Details

The function is used to initialize the processing (folder names, packages, etc.), to launch the GUI (MODIS<sub>tsp</sub>\_GUI) and receive its outputs, and to launch the required routines for downloading and processing the requested datasets.

## Note

License: GPL 3.0



**Author(s)**

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

**See Also**

[MODISrsp\\_GUI](#), [MODISrsp\\_process](#)

**Examples**

```
# Running the tool without any option will start the GUI with the default or last used
# settings
## Not run:
MODISrsp()
## End(Not run)

# Run the tool using the settings previously saved in a specific option file
## Not run:
MODISrsp(gui = FALSE, options_file = "X:/yourpath/youroptions.json")
## End(Not run)

# Run the tool using a previously saved options file,
# but editing it with the GUI before starting the processing
## Not run:
MODISrsp(options_file = "X:/yourpath/youroptions.json")
## End(Not run)

# Run the tool using the settings previously saved in a specific option file
# and specifying the extent from a spatial file
## Not run:
MODISrsp(gui = FALSE, options_file = "X:/yourpath/youroptions.json",
  spatial_file_path = "X:/yourpath/yourspatialfile.shp" )
## End(Not run)

# Run the tool in a batch mode, using the settings previously saved in a specific
# option file and specifying each time the extent from a different spatial file
## Not run:
extent_list = list.files("X:/path/containing/some/shapefiles/", "\\*.shp$")
for (single_shape in extent_list)
  MODISrsp(gui = FALSE, options_file = "X:/yourpath/youroptions.json",
    spatial_file_path = single_shape )
## End(Not run)
```

---

MODISrsp\_addindex

*MODISrsp\_addindex*

---

**Description**

Function used to add a user-defined Spectral Index to the default list of computable spectral indexes. Execution without the GUI (i.e., to add a new index from a script) is also possible (see examples)

**Usage**

```
MODIStsp_addindex(option_jsfile = NA, prodopts_file = NA, selprod = NA,
  selvers = NA, gui = TRUE, new_indexbandname = "",
  new_indexfullname = "", new_indexformula = "",
  new_indexnodata_out = "32767", MODIStsp_dir = NA)
```

**Arguments**

option_jsfile	settings (optional): full path of the JSON file containing the processing options in which the new indexes are saved (default: MODIS <sub>tsp</sub> _Previous.JSON in subfolder Previous).
prodopts_file	settings (optional): full path of the RData file containing products description (default: MODIS <sub>tsp</sub> _ProdOpts.RData in subfolder Previous)
selprod	settings (optional): Name of the product to which the new index should be added. (Note: the index will be added to all other products allowing its computation !)
selvers	settings (optional): Version of the product to which the new index should be added.
gui	logical value (default: TRUE): if TRUE, the GUI is opened to define the new index; otherwise use the "new_indexbandname", "new_indexfullname" and "new_indexformula" parameters to define it non-interactively.
new_indexbandname	(optional if gui=TRUE): short name (acronym) of the new spectral index.
new_indexfullname	(optional if gui=TRUE): extended name of the new spectral index.
new_indexformula	(optional if gui=TRUE): string containing the formula of the new spectral indexes. Variables accepted to compute it are the names of the bands: b1_Red, b2_NIR, b3_Blue, b4_Green, b5_SWIR, b6_SWIR and b7_SWIR.
new_indexnodata_out	(optional): NoData value to assign to the rasters containing the new index
MODIS <sub>tsp</sub> _dir	(optional): main folder containing MODIS <sub>tsp</sub> R files (used only to launch MODIS <sub>tsp</sub> from outside the package using MODIS <sub>tsp</sub> _std.R)

**Details**

- The function asks the user to provide the info related to the new desired Spectral Index using a GUI interface, checks for correctness of provided information (e.g., correct bandnames, computable formula, etc...). If the index is legit, it modifies the MODIS<sub>tsp</sub>\_Previous.RData file so to allow computation of the additional index within MODIS<sub>tsp</sub>./n
- To remove all custom-added spectral indexes, simply delete the MODIS<sub>tsp</sub>\_Previous.RData file within the /Previous subfolder of the folder in which the package was installed, or the alternative JSON specified by the parameter "option\_jsfile".
- The function can be run either from within the main MODIS<sub>tsp</sub> GUI, or a standalone script. In the latter case, it modifies either the MODIS<sub>tsp</sub>\_Previous.RData options file, or the options\_file specified by the user, to add the new index.

**Value**

NULL - the MODIS<sub>tsp</sub>\_Previous.RData file is modified so to allow computation of the additional index

**Note**

License: GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

**Examples**

```
# Run the GUI to interactively define the function
## Not run:
MODIStsp_addindex()
## End(Not run)

# Run the GUI and save the new index in a custom RData file
## Not run:
MODIStsp_addindex(option_jsfile = "X:/yourpath/youroptions.json")
## End(Not run)

# Define the new index non-interactively
## Not run:
MODIStsp_addindex(gui = FALSE, new_indexbandname = "SSD",
  new_indexfullname = "Simple Stupid Difference",
  new_indexformula = "b2_NIR-b1_Red")
## End(Not run)
```

---

MODIS<sub>tsp</sub>\_check\_files    *MODIS<sub>tsp</sub>\_check\_files*

---

**Description**

Accessory function used to see if all expected out files for the selected date are already present.

**Usage**

```
MODIStsp_check_files(out_prod_folder, file_prefix, bandnames,
  bandsel_orig_choice, yy, DOY, out_format, indexes_bandnames, indexes_bandsel,
  quality_bandnames, quality_bandsel)
```

**Arguments**

out_prod_folder	string output folder name
file_prefix	string file prefix of the product (e.g., MOD13Q1)
bandnames	string array Bandnames of the MODIS product
bandsel_orig_choice	0/1 array Indicates which original MODIS layers were selected to be processed
yy	string year
DOY	string doy
out_format	string GTiff or ENVI
indexes_bandnames	string array Name of available spectral indexes for the product
indexes_bandsel	0/1 array Indicates which spectral indexes were selected to be processed
quality_bandnames	string array Name of available Quality Indicators for the product
quality_bandsel	0/1 array Indicates which Quality Indicators were selected to be processed

**Details**

if all expected out files for the selected date are already present, check\_files is set to TRUE, and date is skipped in MODIS<sub>tsp</sub>\_process

**Value**

check - logical = 1 if all expected output files are already existing

**Note**

License: GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

MODISrsp\_extract      *MODISrsp\_extract*

---

### Description

function used to extract time series data from rts files created by MODISrsp on spatial locations provided in the form of "R" spatial objects (spatialPoints, spatialPolygons, etc.)

### Usage

```
MODISrsp_extract(in_rts, sp_object, start_date = NULL, end_date = NULL,
  id_field = NULL, FUN = "mean", out_format = "xts", small = TRUE,
  small_method = "centroids", na.rm = TRUE, verbose = FALSE)
```

### Arguments

in_rts	input "rasterStack" or "rasterBrick" object created by MODISrsp (it MUST contain acquisition dates in the "Z" attribute " )
sp_object	"sp" object OR name of an ESRI shapefile specifying the "positions" from which data has to be extracted If sp_object represents lines, the output object contains one column for each polygon, containing values obtained applying the function specified as the FUN argument over all pixels touched by the line, and one line for each date. If sp_object represents points, the output object contains one column for each point, containing values of the cells corresponding to the pint, and one line for each date. If sp_object represents polygons, the output object contains one column for each polygon, containing values obtained applying the function specified as the FUN argument over all pixels belonging to the polygon, and one line for each date
start_date	"Date", "POSIXct" or "POSIXlt" starting date of the period to be considered for data extraction OR character string coercible to Date class (format = "yyyy-mm-dd"). If not provided, the starting date of the rasterStack is used.
end_date	"Date", "POSIXct" or "POSIXlt" ending date of the period to be considered for data extraction OR character string coercible to Date class (format = "yyyy-mm-dd").If not provided, the ending date of the rasterStack is used.
id_field	character name of the column of the input sp object or shapefile to be used in the data extraction. Values contained in the column MUST be unique The names of the columns of the output are taken from this column. If not provided, or an invalid value is provided, then the names of the columns of the output reflect the number of the feature in the original sp object or shapefile.
FUN	function to summarize the values (e.g. mean) on polygon data frames. The function should take a single numeric vector as argument and return a single value (e.g. mean, min or max), and accept a na.rm argument. Thus, standard R functions not including an na.rm argument must be wrapped as in this example: fun=function(x,...)length(x). Defaults to "mean"
out_format	character "xts" or "dframe". default to 'xts'. If dframe, the output is a data frame with dates in the first column and extracted data in the others

small	logical If set to TRUE, and input is polygons, then values are returned also for polygons not covering at least one raster cell. "Included" cells in this case depend on the values of the "small_method" parameter.
small_method	string 'centroids' or 'full'. if small == T and input is polygons, controls which cells are "extracted" for small polygons. If set to centroids (default), then only the cells corresponding to polygon centroid are considered (faster, may have problems on strangely shaped polygons). If set to "full", then all cells intersected by the small polygon are extracted and used in calculation.
na.rm	Logical If TRUE, and sp_object is a polygon, then na.rm = T is used when applying the function to the different pixels of the polygon. Default = T.
verbose	Logical If TRUE, messages on extraction completion are sent out. Default = T.

### Details

The function takes as input a rasterStack object containing time information in the "z" attribute (set by "raster" function "SetZ"), a starting and ending date and a standard "R" spatial object, and returns the time series for the spatial locations specified in the spatial object in the form of a "R" xts object OR a plain data.frame with a "date" column in first position. If the input spatial object is a "point" or "line" one, the output object contains one column for each specified point, or for each cell intersecting the line, and one line for each date. If the input spatial object is a "polygon" one, the output object contains one column for each polygon, containing values obtained applying the function specified as the FUN argument over all pixels belonging to the polygon, and one line for each date.

### Value

data.frame or xts object. Each column of data corresponds to one point or one polygon

### Note

License: GPL 3.0

### Author(s)

Lorenzo Busetto, PhD (2015) email: busetto.l@irea.cnr.it

### Examples

```
# Extract average and standard deviation values from a rts object created by MODIStsp
# for each polygon of a shapefile, for each date in the period between 2001-01-01 and 2014-12-31
## Not run:
#Set the inputs
infile <- "in_path/MOD13Q1_MYD13Q1_NDVI_49_2000_353_2015_RData.RData" # Input rts file
shpname <- "path_to_file/rois.shp" # Path to Polygon Shapefile
startdate <- as.Date("2010-01-01") # Start date for extraction
enddate <- as.Date("2014-12-31") # End date for extraction
#Load Data
inrts <- get(load(infile)) # Load the rts file
# Compute average and St.dev
dataavg <- MODIStsp_extract(inrts,shpname, startdate, enddate, FUN = 'mean', na.rm = T)
```

```

datasd <- MODISrsp_extract(inrts,shpname, startdate, enddate, FUN = 'sd', na.rm = T)
plot(dataavg)

## End(Not run)

```

---

MODISrsp\_GUI

*MODISrsp\_GUI*


---

## Description

Function used to generate and handle the GUI used to allow selection of MODISrsp processing parameters. If the "previous options" file (MODISrsp\_Previous.json) already exists, it is loaded and used to reinstate the GUI to its last state. Otherwise, the previous options file is created by launching the MODISrsp\_read\_xml function

## Usage

```

MODISrsp_GUI(general_opts, prod_opt_list, scrollWindow, MODISrsp_dir,
             previous_jsfile, prodopts_file)

```

## Arguments

<code>general_opts</code>	data.frame containing general processing options passed by MODISrsp
<code>prod_opt_list</code>	List of MODIS products specifications (read from MODISrsp_ProdOpts.xml file)
<code>scrollWindow</code>	logical parameter passed by MODISrsp main function.
<code>MODISrsp_dir</code>	main folder of the package
<code>previous_jsfile</code>	json parameters file containing data of the last execution
<code>prodopts_file</code>	character name of RData file containing info about MODIS products

## Value

Quit - Logical - tells the main if running processing or exiting (also, Processing options are saved in "previous" file and (if "Save options" is pressed) in user's selected file)

## Note

License: GPL 3.0

## Author(s)

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

MODIS<sub>tsp</sub>\_process      *MODIS<sub>tsp</sub>\_process*

---

### Description

main function of MODIS<sub>tsp</sub> tool. Takes as input processing parameters specified by the user using MODIS<sub>tsp</sub>\_GUI and saved in MODIS<sub>tsp</sub>\_Previous.json (Interactive use), or a user specified JSON file (batch use) (See MODIS<sub>tsp</sub>\_main for details ) and performs all required processing.

### Usage

```
MODIStsp_process(sel_prod, start_date, end_date, out_folder, out_folder_mod,
  reprocess = "Yes", delete_hdf = "No", sensor, download_server, user,
  password, https, ftps, start_x, start_y, end_x, end_y, bbox, out_format,
  compress, out_res_sel, out_res, native_res, tiled, MOD_proj_str, outproj_str,
  nodata_in, nodata_out, nodata_change, scale_val, scale_factor, offset, rts,
  datatype, bandsel, bandnames, indexes_bandsel, indexes_bandnames,
  indexes_formula, indexes_nodata_out, quality_bandnames, quality_bandsel,
  quality_bitN, quality_source, quality_nodata_in, full_ext, quality_nodata_out,
  file_prefixes, main_out_folder, resampling, ts_format, use_aria = TRUE,
  download_range = "full", gui = TRUE)
```

### Arguments

sel_prod	string selected MODIS product
start_date	string start_date for images download and preprocessing (yyyy.mm.dd)
end_date	string end_date for images download and preprocessing (yyyy.mm.dd)
out_folder	main output folder
out_folder_mod	output folder for original HDF storage
reprocess	string string ("Yes"/"No") If Yes, reprocess data for already existing dates (Default = 'Yes')
delete_hdf	string ("Yes"/"No") If Yes, delete original HDF after completion
sensor	string ("Terra" or "Aqua" or "Both")
download_server	service used to download MODIS tiles, one of: 'http', 'ftp', NA.
user	Username for http download ( <a href="https://urs.earthdata.nasa.gov/home">urs.earthdata.nasa.gov/home</a> )
password	Password for http download ( <a href="https://urs.earthdata.nasa.gov/home">urs.earthdata.nasa.gov/home</a> )
https	hash https site for download of HDF of selected product
ftps	hash ftps site for download of HDF of selected product
start_x	int start horizontal tile
start_y	int start vertical tile
end_x	int end horizontal tile



end_y	int end vertical tile
bbox	array output bounding box (xmin, xmax, ymin, ymax ) in out proj coords
out_format	string output raster format (ENVI or GTiff)
compress	string compression for GTiff outputs (None, LZW, DEFLATE)
out_res_sel	string "Native" or "Resampled"
out_res	float Output resolution (in output projection measurement unit)
native_res	float Native resolution of MODIS product
tiled	0/1 1 = tiled product; 0 = non-tiled product (resolution 0.05 deg)
MOD_proj_str	string proj4 string for MODIS product native projection
outproj_str	string proj4 string of selected output projection
nodata_in	array Original NoData for MODIS bands
nodata_out	Target NoData for MODIS bands
nodata_change	string (Yes/No) if Yes, NoData are set to nodata_out in output rasters
scale_val	string (Yes/No) if Yes, output values in are rescaled in the measure unit of the variable
scale_factor	scale factor to be applied to MODIS layer to convert from scaled integer to correct measure units
offset	offset to be applied to MODIS layer to convert form scaled integer to correct measure units
rts	string ("Yes"/"No") If Yes, create rts time series
datatype	string array datatypes of MODIS bands
bandsel	array of length equal to number of original modis layers. set to 1 for bands to be processed
bandnames	array of Abbreviated Names of MODIS bands
indexes_bandsel	array of length equal to number of available spectral indexes, set to 1 for indexes to be processed
indexes_bandnames	array of Abbreviated Names of MODIS indexes
indexes_formula	array of indexes formulas
indexes_nodata_out	NoData values for indexes
quality_bandnames	array of Names of MODIS quality indicators
quality_bandsel	array of length equal to number of available quality indicators, set to 1 for indicators to be processed
quality_bitN	list of strings with number of entries equal to number of quality indicators. each entry contains position of bits corresponding to a QI (e.g., 0-1)

quality_source	list of strings which connects each quality indicator to its source aggregated quality assurance layer
quality_nodata_in	Always 255
full_ext	string ("Full_Ext" or "Resized")
quality_nodata_out	Always 255
file_prefixes	output file prefix according to selected product (e.g., MOD13Q1)
main_out_folder	Suffix to add to the overall out_folder to create the out dir for the product (corresponds to an abbreviation of the selected product)
resampling	string resampling method (near, bilinear, etc.)
ts_format	string format of virtual files (None, ENVI Meta Files, GDAL vrt files, ENVI and GDAL)
use_aria	logical if TRUE, then aria2c is used to accelerate download (if available !)
download_range	character if "full", all the available images between the starting and the ending dates are downloaded; if "seasonal", only the images included in the season (e.g: if the starting date is 2005-12-01 and the ending is 2010-02-31, only the images of December, January and February from 2005 to 2010 <ul style="list-style-type: none"> <li>• excluding 2005-01, 2005-02 and 2010-12 - are downloaded)</li> </ul>
gui	logical indicates if processing was called within the GUI environment or not. If not, direct processing messages to the log

## Details

After retrieving the input processing options, the function

1. Accesses lpdac http or ftp archive to determine the list of dates to be processed
2. Performs all required processing steps on each date (download, reprojection, resize, mosaicing, indexes computation, quality indicators computation)
3. Creates virtual files of the processed time series.

Reprojection and resize is dealt with by accessing gdal routines through the gdaUtils package. Extraction of bitfields from Quality layers is done through fast bitwise computation Checks are done in order to not re-download already existing HDF images, and not reprocess already processed dates (if the user did not specify that)

## Note

Thanks Tomislav Hengl and Babak Naimi, whose scripts made the starting point for development of this function ([ModisDownload](#); [Download\\_and\\_resampling\\_of\\_MODIS\\_images](#))

License: GPL 3.0

## Author(s)

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

 MODISrsp\_process\_indexes

*MODISrsp\_process\_indexes*


---

### Description

function used to compute spectral indexes, given the index formula

### Usage

```
MODISrsp_process_indexes(out_filename, formula, bandnames, nodata_out,
  out_prod_folder, indexes_nodata_out, file_prefix, compress, yy, DOY,
  out_format, scale_val)
```

### Arguments

out_filename	character	basename of the file in to which save results
formula	character	Index formula, as derived from XML file and stored in prod_opts within previous_file
bandnames	character	array of names of original HDF layer. Used to identify the bands required for index computation
nodata_out	character	array of NoData values of reflectance bands
out_prod_folder	character	output folder for the product used to retrieve filenames of rasters of original bands to be used in computations
indexes_nodata_out	character	NoData value for resulting raster
file_prefix	character	used to retrieve filenames of rasters of original bands to be used in computations
compress	character	compression option for GTiff files
yy	character	year string used to retrieve filenames of rasters of original bands to be used in computations
DOY	character	doy string used to retrieve filenames of rasters of original bands to be used in computations
out_format	character	string used to retrieve filenames of rasters of original bands to be used in computations
scale_val	character	(Yes/No) if Yes, output values in are computed as float -1 - 1, otherwise integer -10000 - 10000

### Details

the function parses the index formula to identify the required bands. On the basis of identified bands, it retrieves the reflectance bands required, gets the data into R raster objects, performs the computation and stores results in a GeoTiff or ENVI raster file

**Value**

NULL - new raster file saved in out\_filename

**Note**

License: GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2017) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2017) <ranghetti.l@irea.cnr.it>

---

MODIS<sub>tsp</sub>\_process\_QA\_bits

*compute Quality Indicators from HDF bit-field layers*

---

**Description**

function used to extract quality indicator from MODIS aggregated quality layers

**Usage**

```
MODIStsp_process_QA_bits(out_filename, in_source_file, bitN, out_format,
    nodata_source, nodata_qa_in, nodata_qa_out, compress)
```

**Arguments**

out_filename	character file name of the output raster files containing QI values
in_source_file	character name of the file created by MODIS <sub>tsp</sub> containing the data required to compute the quality indicator
bitN	character position of the bits corresponding to the quality indicator of interest (e.g., 0-1 = first two bits; 2-5: bits from 2 to 5, etc.)
out_format	output format (ENVI or GTiff)
nodata_source	character NoData values of the MODIS band containing data from which the bit field corresponding to the quality indicator must be extracted
nodata_qa_in	character in NoData for quality bands ("255")
nodata_qa_out	character out NoData for quality bands ("255")
compress	character compression option for GTiff files

**Details**

On the basis of the name of the image containing the aggregated quality information (in\_source\_file``) and of the position of the bit field, the function extracts the correct information exploiting bitwise operators, and save the result in a new raster image

**Note**

License: GPL 3.0 Based on the "modis.qc.R" script by Yann Chemin (2008) (<https://goo.gl/7Fhreo>)  
license GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2017) <busetto.l@irea.cnr.it>  
Luigi Ranghetti, PhD (2017) <ranghetti.l@irea.cnr.it>

---

MODIS<sub>tsp\_read\_xml</sub>      *MODIS<sub>tsp\_read\_xml</sub>*

---

**Description**

function used to parse the XML file used to store the characteristics of MODIS Land Products and store them in the "prod\_opts" data frame

**Usage**

```
MODIStsp_read_xml(prodopts_file = prodopts_file, xml_file = xml_file)
```

**Arguments**

prodopts\_file    string filename of the RData in which to store the data parsed from the XML file  
xml\_file        string filename of the XML file containing the MODIS products characteristics

**Details**

The function parses the XML file product by product, stores data in a data frame and saves the data frame within the "MODIS<sub>tsp\_previous</sub>" RData file as a list of lists

**Value**

NULL - retrieved data are stored in the specified RData file

**Note**

License: GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>  
Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

MODISvsp\_vrt\_create     *Create MODISvsp virtual files*

---

### Description

Function used to create virtual files from time series of single-band files corresponding to different acquisition dates

### Usage

```
MODISvsp_vrt_create(out_prod_folder, meta_band, file_prefixes, sens_sel,
                    ts_format, nodata_value, out_format, rts)
```

### Arguments

out_prod_folder	string main output folder
meta_band	string "name" of the band (or SI, or QI) for which the virtual file is to be created
file_prefixes	string file_prefixes for TERRA and AQUA - used to identify the files corresponding to each sensor
sens_sel	string name of the sensor for which the time series has to be created (Aqua, Terra, Mixed) If "Mixed" and both TERRA and AQUA images are available, a "mixed" virtual file comprising data from both sensors ordered on DOY base is created
ts_format	string required output format for virtual file (ENVI, GDAL, Both)
nodata_value	string NoData value to be used for vrt files (equal to NoData value of inputs)
out_format	format of images used as "input" for the vrt and contained in out_prod_folder/band folders (ENVI or GTiff)
rts	string ("Yes"/"No") If Yes, create rts time series

### Details

The function takes as input the folder in which the single-band files are stored, and creates a ENVI Meta file and/or a GDAL vrt file that allows access to the full time series as if it was a single physical file

### Value

NULL - virtual files are stored in the "Time Series" subfolder of out\_prod\_folder

### Note

License: GPL 3.0

**Author(s)**

Lorenzo Busetto, PhD (2014-2015) <busetto.l@irea.cnr.it>

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

---

reproj\_bbox

*reproj\_bbox*

---

**Description**

Ancillary function used to reproject bounding boxes; setting the parameter 'enlarge' allows to choose if the new one would be the one which completely include the old or if is simply the one obtained by reprojecting the upper-left and the lower-right corners.

**Usage**

```
reproj_bbox(bbox, in_proj, out_proj, enlarge = TRUE, N_dens = 1000)
```

**Arguments**

bbox	The input bounding box (it can be a matrix in the bbox format or a vector in the format (xmin, ymin, xmax, ymax).
in_proj	The input projection (string format).
out_proj	The output projection (string format).
enlarge	Logical parameters: if TRUE, the reprojected bounding box is the one which completely include the original one; if FALSE, it is simply the one obtained by reprojecting the upper-left and the lower-right corners.
N_dens	Densification ratio used in the case enlarge is TRUE.

**Note**

License: GPL 3.0

**Author(s)**

Luigi Ranghetti, PhD (2015) <ranghetti.l@irea.cnr.it>

# Index

[bbox\\_from\\_file](#), 2

[install\\_MODISrsp\\_launcher](#), 3

[lpdaac\\_getmod\\_dates](#), 4

[lpdaac\\_getmod\\_dirs](#), 5

[lpdaac\\_getmod\\_names](#), 6

[MODISrst\\_check\\_md5](#), 7

[MODISrsp](#), 8

[MODISrsp\\_addindex](#), 9

[MODISrsp\\_check\\_files](#), 11

[MODISrsp\\_extract](#), 13

[MODISrsp\\_GUI](#), 9, 15

[MODISrsp\\_process](#), 9, 16

[MODISrsp\\_process\\_indexes](#), 19

[MODISrsp\\_process\\_QA\\_bits](#), 20

[MODISrsp\\_read\\_xml](#), 21

[MODISrsp\\_vrt\\_create](#), 22

[reproj\\_bbox](#), 23