MODIStsp: An R package for automatic preprocessing of MODIS time series - Supplementary Materials

Institute for Electromagnetic Sensing of Environment (IREA-CNR)–Via Bassini 15, Milano (IT)

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

1 Installation Instructions

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.

1.1 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![1]

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")
```

1.2 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 libcurl4-openssl-dev libgdal-dev libproj-dev
```

---

[1]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!

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):

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

2 Running the tool

2.1 Interactive mode

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

```
library(MODIStsp)
MODIStsp()
```

This opens a GUI from which processing options can be specified and eventually saved (or loaded). A description of the different processing parameters to be selected is reported in section 2.1.1 of the main manuscript.[2]

2.2 Non-Interactive mode (Using a previously saved options file)

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 from the GUI
options_file = "X:/yourpath/youroptions.RData"
MODIStsp(gui = FALSE, options_File = options_file)
```

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("X:/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 spatial file).
```

---

[2]At the first execution of *MODIStsp*, a Welcome screen will appear, signaling that MODIStsp 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), MODIStsp (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: 1. Ensure that GDAL is properly installed in your system 2. (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) 3. If nothing works, signal it in the issues GitHub page of MODIStsp and we'll try to help!

```
for (single_shape in extent_list) {
  MODIStsp(gui = FALSE, options_File = "X:/yourpath/youroptions.RData",
          spatial_file_path = single_shape )
}
```

3 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:** `yourpath_to_MODIStsp_sh/MODIStsp.sh -g -s "/yourpath/youroptions.RData"`

(launch `yourpath_to_MODIStsp_sh/MODIStsp.sh -h` for details).

- **In Windows:** `yourpath_to_MODIStsp_bat\MODIStsp.bat -g -s "X:/yourpath/youroptions.RData"`

(launch `yourpath_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 typing

     ```
     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 /yourpath_to_MODIStsp_sh/MODIStsp.sh -g -s "/yourpath/youroptions.RData"
     ```

   - **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.RData"` as argument.

3.1 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.RData"`.

- **In Windows::** this creates a link in the Start Menu and optionally a desktop shortcut.

See `?install_MODIStsp_launcher` for details and path customisations.

4 Installing R and GDAL

4.1 Installing R

4.1.1 Windows

Download and install the latest version of R which can be found here.

4.1.2 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 (altough recommended), since packaged version of R is $\geq$ 3.2.1 (although not the latest); in this case, user can insall R by simply typing in a terminal

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

4.2 Installing GDAL $\geq$ 1.11.1

4.2.1 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

4.2.2   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, altough recommended in order to have updated version of GDAL installed.

2. To install GDAL a terminal and type

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

4.2.3   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.

4.2.4   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).