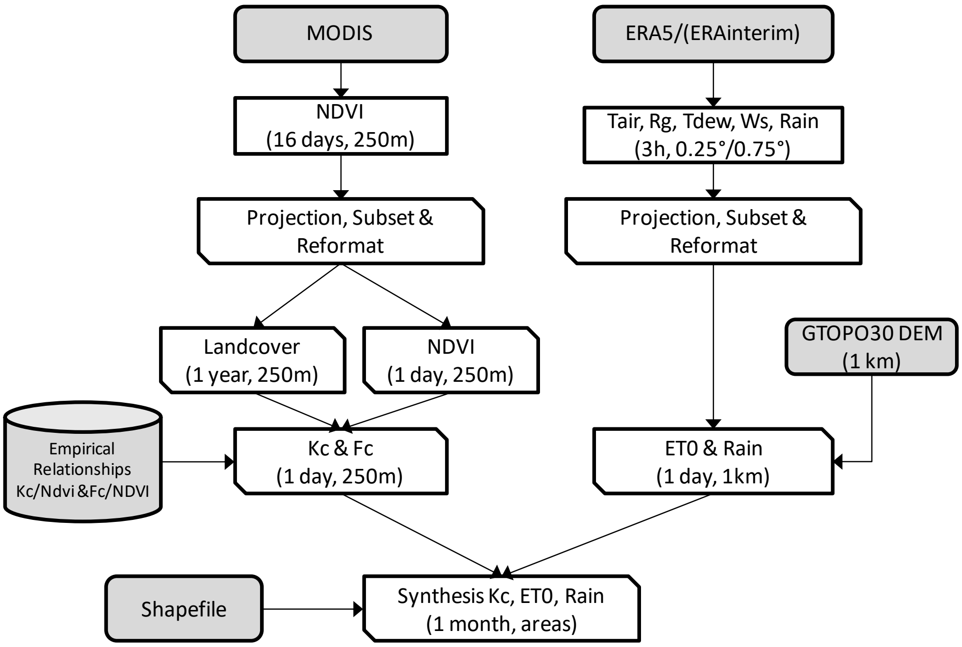
# mET-weap

# Installation Procedure & User guide

mET-weap is a python package that process MODIS-NDVI and ECMWF meteorological time series to produce input data for WEAP21: Reference Evapotranspiration (ET0), Precipitation, Temperature, Wind Speed, Crop coefficient (Kc) and Fraction Cover (Fc). Those datasets are synthesized for the user polygons and at a monthly or daily time step.


MODIS
ERA5/(ERAinterim)
NDVI (16 days, 250m)
Tair, Rg, Tdew, Ws, Rain (3h, 0.25°/0.75°)
Projection, Subset & Reformat
Projection, Subset & Reformat
Landcover (1 year, 250m)
NDVI (1 day, 250m)
GTOPO30 DEM (1 km)
Empirical Relationships Kc/Ndvi &Fc/NDVI
Kc & Fc (1 day, 250m)
ET0 & Rain (1 day, 1km)
Shapefile
Synthesis Kc, ET0, Rain (1 month, areas)

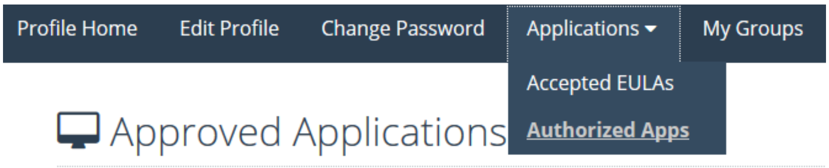
**Funding**: The development was funded by the AMETHYST project (ANR-12-TMED-0006-01, ANR TRANSMED program) and the LMI TREMA.

**Authors**: Michel Le Page wrote the main parts of the code. Cindy Gosset worked on several developments and testing around the ERAinterim dataset and Antoine Beaumont worked on a distributable version (packaging, license).

# 1- REQUIREMENTS

## 1) Subscribe to Earthdata

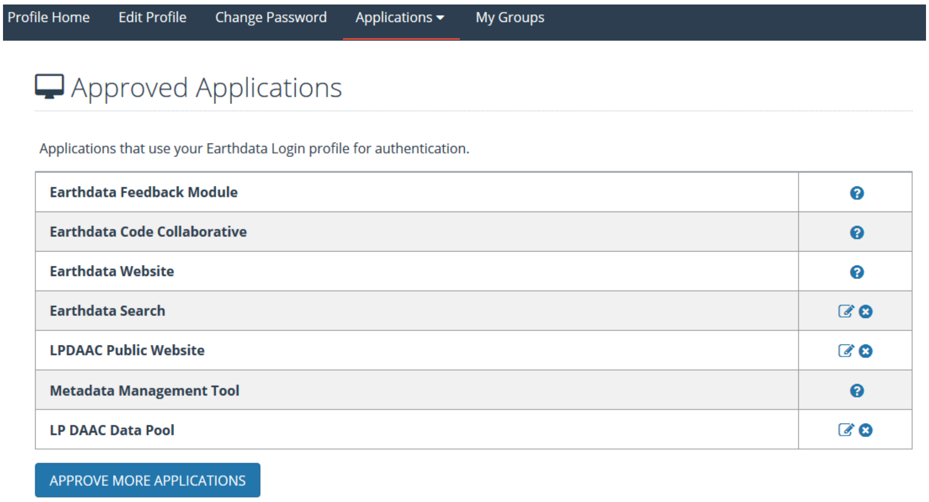
Create an account on https://urs.earthdata.nasa.gov, and then go to “Applications->Authorized Apps”:



In “Approve more applications”:

- Approve “LP DAAC Data Pool” and “Earthdata Search” applications for the MODIS data.

- Approve “NASA GESDISC DATA ARCHIVE” for the MERRA2 data.



Applications that use your Earthdata Login profile for authentication.

| | |
|---|---|
| Earthdata Feedback Module | ? |
| Earthdata Code Collaborative | ? |
| Earthdata Website | ? |
| Earthdata Search | ✎ ⊗ |
| LPDAAC Public Website | ✎ ⊗ |
| Metadata Management Tool | ? |
| LP DAAC Data Pool | ✎ ⊗ |

## 2) Subscribe to CDS (Climate Data Store)

Click on login/register on this website: https://cds.climate.copernicus.eu/#!/home

When your account is created, accept the license Copernicus product on CDS API: https://cds.climate.copernicus.eu/cdsapp/#!/terms/licence-to-use-copernicus-products

Then, click on your name near the logout button to retrieve your API key on your user profile.

Copy and paste your API key into the userconfig.py file in the mET_weap_test folder (see I) 1.3) paragraph). Use the format UID:API key.

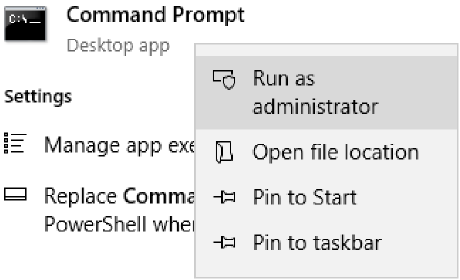
## 3) Install Gdal binaries (Windows)

Go to https://trac.osgeo.org/osgeo4w/ and click on the 32bit.exe or 64bit.exe depending on your computer system.

Execute the osgeo4w.exe and follow the default installation.

In order to finalize the installation; you must add an environmental variable to Windows.

- Open the command prompt as an administrator



- Execute: setx GDAL_DATA “C:\OSGeo4W64\share\gdal” /M

## 4) Required files

### Geopotential_cdsapi

This file is downloaded in the root directory. For example, “C:\\HAOUZ” in the test.

### GTOPO30

For the DEM module, a GTOPO30 file is required. Follow these steps to download it:

- Connect to the website : https://earthexplorer.usgs.gov/
- Click on login and create new account
- Click on the confirmation link received by email
- Go on “Search Criteria” and click on the map with the left mouse button to define a landmark in the middle of your studied area
- Click on “results”
- Then click on “Download options” and download the geoTIFF file
- Put the geoTIFF file in the repertory mET_weap_test\Gtopo30 (see paragraph I)1.3) )

### Zoning

For the spatio-temporal synthesis module, an ESRI shapefile is required.  The shapefile must be a single part shapefile and must contain at least one identification attribute.

The path and key attribute are indicated in the configuration file.

# 2- Installation of mET_weap

## 1) Python distribution

The mET_weap package uses the Anaconda distribution (https://www.anaconda.com) and has been tested with Python 3.3, 3.4 and 3.7.

When the installation of Anaconda is done, the mET_weap package can be installed on the base environment.

## 2) Installation

### From the source folder

Download the mET_weap package from Github. In this directory, mET_weap is the source folder of the setup.py and mET_weap_test is the test folder (see paragraph I)1.3) ).

Copy/Paste this source folder “mET_weap” on your computer.

Then, in the anaconda prompt, execute:

- cd “source folder of the setup.py”
- For example: cd "C:\Users\VI\Documents\mET_weap"
- python setup.py sdist bdist_wheel
- cd .\dist
- pip install mET_weap-1.3.tar.gz
- conda install gdal

To verify that the package is well installed, check if the two folders “mET_weap” and “mET_weap-0.1.dist-info” are located in the “sites-packages” folder.

(For example: C:\Users\VI\Anaconda3\Lib\site-packages\mET_weap)

### From the PyPI (Python Package Index):

In the anaconda prompt, execute:

- pip install mET_weap

### 3) Test folder

Recover the “mET_weap_test” folder in the package downloaded on Pypi or Github.
This folder can be also recovered in the “sites-packages” folder (C:\Users\VI\Anaconda3\Lib\site packages\mET_weap for instance) if the installation is made directly from the Pypi.
This folder allows to test the program and better understand its functioning.
Copy/paste “mET_weap_test” on your desktop to start the test.

## 3- Short scripts description

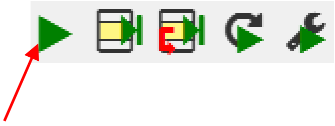
### userconfig.py

See the README.md

### usermain.py

main.py is the manager code

You can for example run it in Spyder.



Or run it on the Anaconda prompt:

- Execute “cd (root directory)” ( cd C:\Users\VI\Documents\mET_weap)
- Execute “python usermain.py”

### download_geopotential_cdsapi.py

The ‘download_geopotential_cdsapi.py’ script allows to download the ‘geopotential_cdsapi.nc’ file in the 'C:\\HAOUZ’ directory. This file is required for the DEM extraction.

### DEM_1km.py

The “DEM_1km.py” script allows to:

- Subset the DEM from the geopotential and the local topography.
- Rescale the DEM at the chosen resolution (1km by default)
- Compute the difference Dz.

The float echelle is the scale at which the Meteo Data will be processed compared to the Modis NDVI dataset (1: 250M, 0.25 : 1000m).

‘DEM_cdsapi’ is extracted from the Gtopo30 file.
(‘mET_weap_test\\Gtopo30\\gt30w020n40.tif’). ‘DEM_ERA_cdsapi’ is extracted from the ‘geopotential_cdsapi.nc’.

## get_modis.py

Get_modis.py is a code developed by J Gomez-Dans j.gomez-dans@ucl.ac.uk, and distributed under licence GPLv3 (http://github.com/jgomezdans/get_modis/).

The program allows downloading MODIS data from the USGS website using the HTTP transport. This program is able to download daily, monthly, 8-daily, etc products for a given year, it only requires the product names (including the collection number), the year, the MODIS reference tile and additionally, where to save the data to, and whether to verbose. The user may also select a temporal period in terms of days of year. Note that as of summer 2016, NASA requires that all downloads are identified with a username and password.

In mET_weap, we use this script to download MOD13Q1 time series.

## download_ERA5_cds.py and ecmwf_daily.api

download_ERA5_cds() is the function to download data from the corresponding server.

The script downloads the following variables:

- 10m_u_component_of_wind
- 10m_v_component_of_wind
- 2m_dewpoint_temperature
- 2m_temperature
- surface_pressure
- surface_solar_radiation_downwards
- total_precipitation

The ERA5.nc outputs files are downloaded in the path ‘ERA5’ folder.

('C:\\HAOUZ\\SAMIR\\data\\ERA5' in the test)

## hdf2tif.py

The “hdf2tif.py” script allows to:

- Extract one subdataset, project it to WGS84 and convert it from HDF to TIF
- subset the image to the given window

The “250m16daysNDVI” outputs are computed in the path 'MOD13Q1_tif' folder.

('C:\\HAOUZ\\SAMIR\\tif\\MOD13Q1' in the test folder).

## Classification.py

The “classification” script determines a yearly land use of the studied area thanks to the “250m16daysNDVI” obtained in the projection and subset module.

The land use is defined in five classes depending on NDVI thresholds: annual culture, self-propagating plant, bare ground, tree and tree on vegetation.

The yearly classifications are computed in the path ‘outputs’ folder. (class_2015.tif and class_2016.tif in 'C:\\HAOUZ\\SAMIR\\tif\\outputs_ERA5' in the test folder).

## Kc_Fc.py

Kc_Fc.py computes the Fc and Kc parameters for every NDVI image according to the empirical relationship chosen for each landcover and specified in the configuration file.

The Kc and Fc image is computed in the path 'outputs_ERA5' folder. ('C:\\HAOUZ\\SAMIR\\tif\\outputs_ERA5' in the test).

## ecmwf_daily.py/era5_daily()

The era5_daily() function from the "ecmwf_daily.py" script computes the daily ET0, precipitation, temperature and average wind speed from the ERA5 tif dataset.

The outputs files are stored under the path 'ERA5_tif'.

( 'C:\\HAOUZ\\SAMIR\\tif\\ERA5' in the test).

## Kc_Fc.py/ET_actual()

The function ET_actual() from the "Kc_Fc.py" interpolate crop coefficients (Kc) and fraction covers (Fc) to a daily time step and compute ETc=ET0*Kc.

The daily outputs are computed in the path 'outputs'.

('C:\\HAOUZ\\SAMIR\\tif\\outputs_ERA5' in the test).

## synthesize.py

Synthesize.py takes a time series of daily input images and computes a spatio-temporal synthesis for the user polygons. The principle is to vectorize the input raster, combine it with the input administrative vectors and compute the proportion of each pixel belonging to the administrative area. To improve calculation time, the result of crossing vectors and weights are obtained only once, and saved for further calculations.

The process is separated into four parts:

0) Init
1) Spatial Cross Shapefile with Raster File
2) Synthesis of each input file according to the proratas computed in part2
3) Temporal Synthesis and writing of CSV files

The outputs are:

A csv files in Weap21 format computed in the path 'synthesis'.

( 'C:\HAOUZ\SAMIR\synthesis_ERA5' in the test).

a vectorized shapefile fishnet computed in the path 'temp'.

( 'C:\HAOUZ\SAMIR\fishnet_ERA5' in the test).

# References

Le Page, M., Y. Fakir, F. Molle, A. Boone, B. Berjamy, L. Jarlan, M. Montginoul, S. Khabba, M. Zribi, Using remote sensing time series to build quantitative irrigation scenarios under climate change (2018), Regional Environmental Change, under review

Le Page, Michel, B. Berjamy, Y. Fakir, F. Bourgin, L. Jarlan, a. Abourida, M. Benrhanem, et al. « An Integrated DSS for Groundwater Management Based on Remote Sensing. The Case of a Semi-arid Aquifer in Morocco ». Water Resources Management 26, no 11 (juin 2012): 3209–3230. https://doi.org/10.1007/s11269-012-0068-3.

Simonneaux, Vincent, Michel Le Page, Dorian Helson, et Sonia Thomas. « Estimation spatialisée de l'évapotranspiration des cultures irriguées par télédétection: Application à la gestion de l'irrigation dans la plaine du Haouz ». Secheresse 20, no 1 (2009): 123–130. https://doi.org/10.1684/sec_2009.0177.

Er-Raki, S., a. Chehbouni, N. Guemouria, B. Duchemin, J. Ezzahar, et R. Hadria. « Combining FAO-56 model and ground-based remote sensing to estimate water consumptions of wheat crops in a semi-arid region ». Agricultural Water Management 87, no 1 (janvier 2007): 41–54. https://doi.org/10.1016/j.agwat.2006.02.004.

Duchemin, B., R. Hadria, S. Er-Raki, G. Boulet, P. Maisongrande, A. Chehbouni, R. Escadafal, et al. « Monitoring wheat phenology and irrigation in Central Morocco: On the use of relationships between evapotranspiration, crops coefficients, leaf area index and remotely-sensed vegetation indices ». Agricultural Water Management 79, no 1 (janvier 2006): 1–27. https://doi.org/10.1016/j.agwat.2005.02.013.