PySEBAL Documentation

Release 0.2

Sajid Pareeth

Dec 09, 2021

Contents:

CHAPTER 1

PySEBAL Installation

PySEBAL is a python library to compute Actual EvapoTranspiration (ETa) and other related variables using SEBAL model. Following specifications are recommended to run PySEBAL.

1.1 Computing requirements

1.1.1 Hardware

- CPU with 2 cores and > 2GHz processor
- Minimum of 8 GB RAM
- Storage space of 10 - 20 GB if processing multiple landsat tiles

1.1.2 Operating systems

- Windows 7/10 (Windows 8, 8.1 should also work, provided dependencies are met)
- Linux (Tested in Ubuntu 18.04 LTS, other Linux OS should also work)

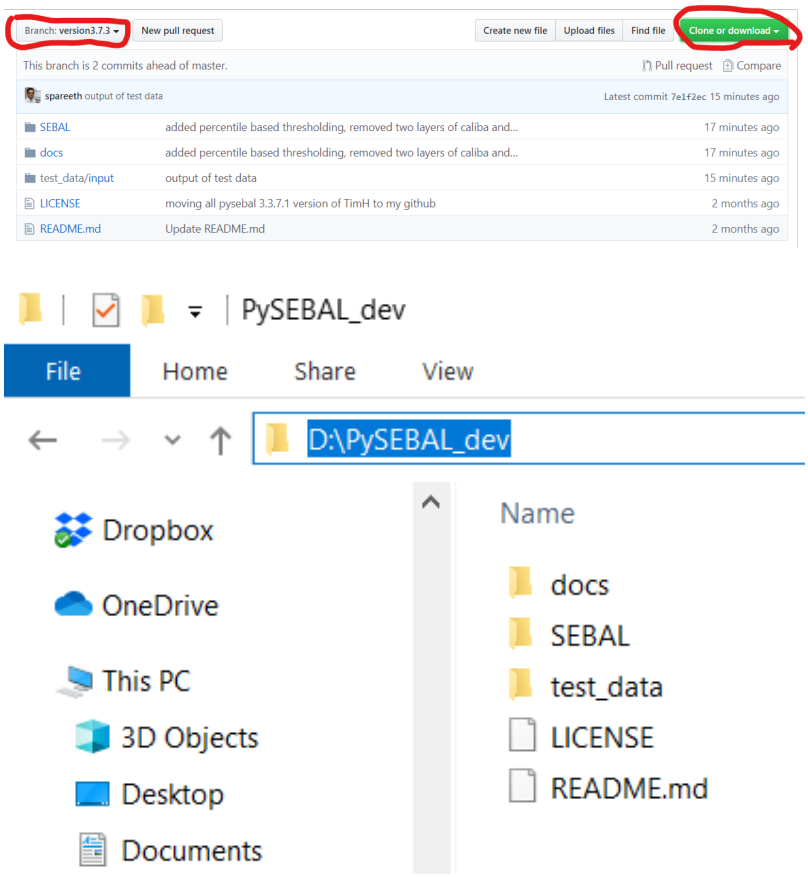
PySEBAL requires Python 3, (tested in python > 3.6).

1.2 Source code

The PySEBAL library is hosted in a publicly available github repository. The library can be downloaded from here. In the link select the latest `version3.7.3` and **Download ZIP**.

Once it is downloaded, unzip it (use *extract here*) into your `D:\` drive or any drive other than `C:\` drive. Rename the folder `PySEBAL_dev-version3.7.3` to `PySEBAL_dev`.

The directory structure after download and unzip should like like below.



1.3 Installation in Windows

The PySEBAL library has multiple dependencies to support spatial data processing and computing. All the required libraries are open source. We reccomend using the OSGeo4W installer and environment to install all the dependencies and to run PySEBAL library in command line.

1.3.1 Installing dependencies

Following steps explain the installation procedure: **Step1 - Download the OSGeo4W installer**

Get the OSGeo4W installer from this link .

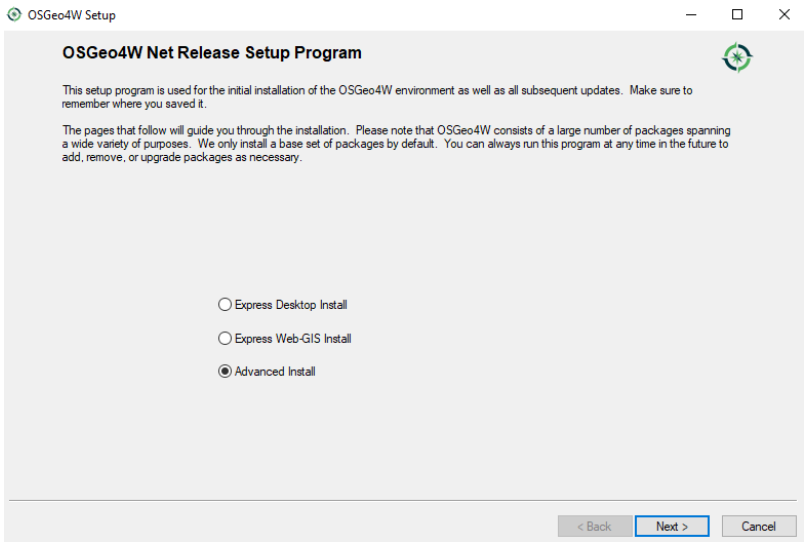
Step2 - Install the dependencies

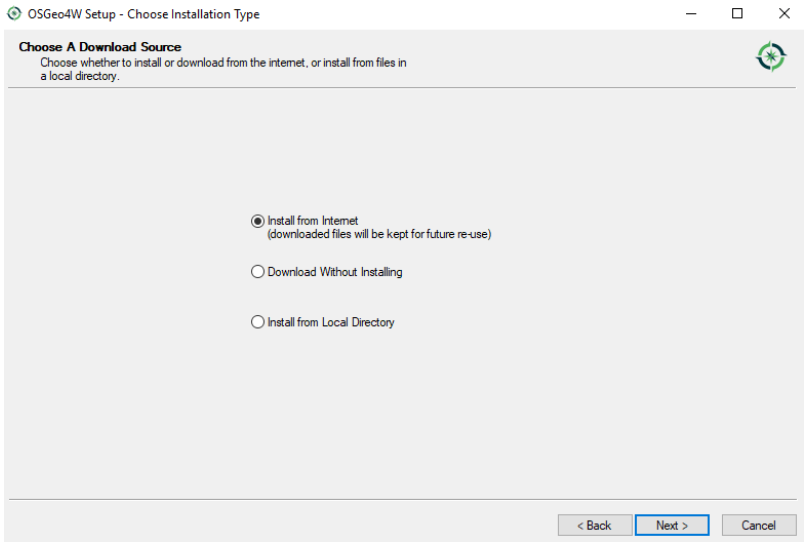
Double click the OSGeo4W installer

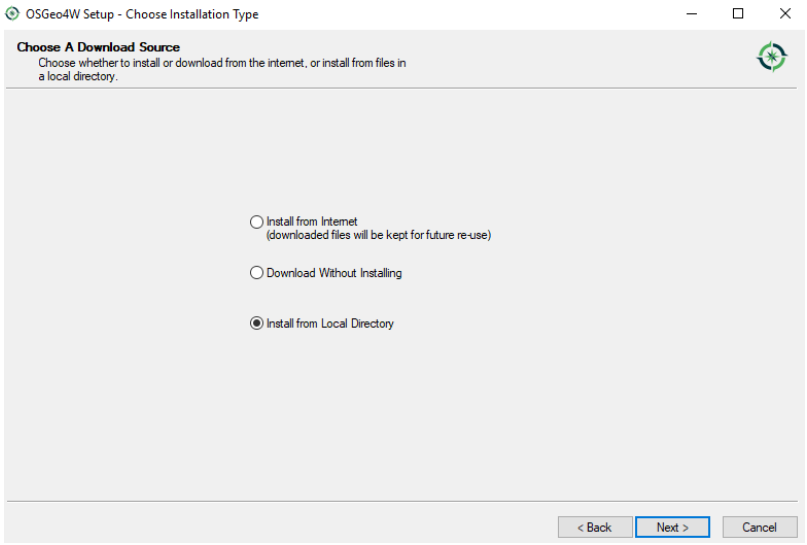
- Select “advanced install” and click “Next”
- In this step there are two options, choose option 1 OR 2. Most cases option 1, unless you get a package of source libraries during a training.

    1. Select “Install from internet” and click “Next”, you must be connected to a good internet.

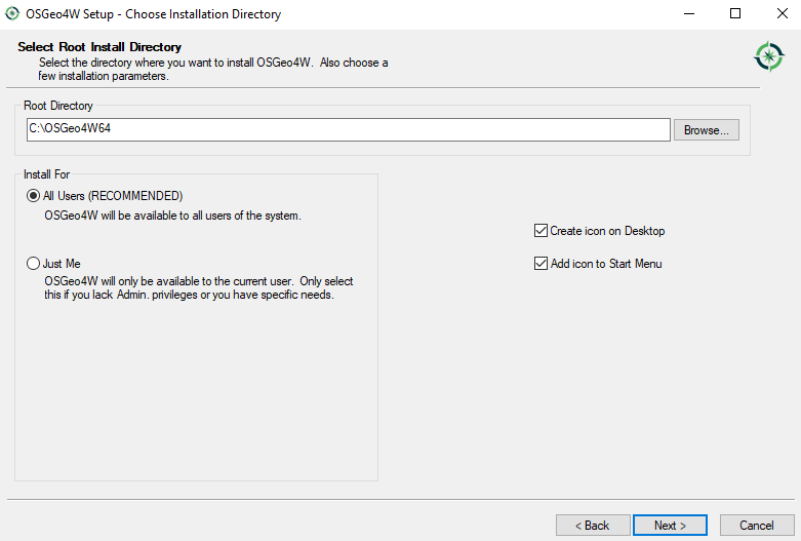
    2. Select “Install from Local Directory” and click “Next”, if you want to install from the source libraries provided to you in USB.


OSGeo4W Setup
OSGeo4W Net Release Setup Program
This setup program is used for the initial installation of the OSGeo4W environment as well as all subsequent updates. Make sure to remember where you saved it.
The pages that follow will guide you through the installation. Please note that OSGeo4W consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.
Express Desktop Install
Express Web-GIS Install
Advanced Install
< Back
Next >
Cancel

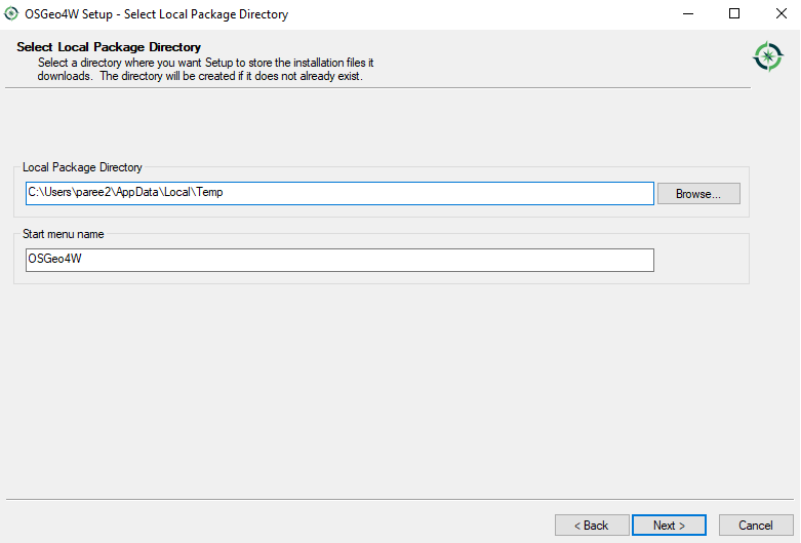

OSGeo4W Setup - Choose Installation Type
Choose A Download Source
Choose whether to install or download from the internet, or install from files in a local directory.
Install from Internet
(downloaded files will be kept for future re-use)
Download Without Installing
Install from Local Directory
< Back
Next >
Cancel


OSGeo4W Setup - Choose Installation Type
Choose A Download Source
Choose whether to install or download from the internet, or install from files in a local directory.
Install from Internet
(downloaded files will be kept for future re-use)
Download Without Installing
Install from Local Directory
< Back
Next >
Cancel

- In this step select the root directory and access to users, keep default settings, and optionally “Create icon on Desktop” for easy access.



- Here choose the folder with local repository (provided to you in USB) if you have selected option 1 in the previous step **or** choose the folder (default option) to download the libraries from internet if you have selected option 2 in the previous step and click “Next”.



- In case of option 2 “Install from internet” in the previous step, select the default option “Direct Connection” and click “Next”.
- In case of option 2 “Install from internet” in the previous step, select the default option “http://download.osgeo.org” as the download site and click “Next”.
- In this step, search for the following packages **one by one**, and select the appropriate (latest) versions by clicking the icon under the column **New**. Check under the **Package** column if you are selecting exact library as stated below.

Warning: Do not click next before selecting all the packages listed below !

The required libraries are:

- qgis-ltr
- grass (select version 7.8.5-1)
- qgis-ltr-grass-plugin7
- msys
- pyproj (select python3-pyproj)
- pandas (select both python3-pandas and python3-geopandas packages)
- scipy (select python3-scipy)
- tcltk (select python3-tcltk)
- pip (select python3-pip)
- setuptools (select python3-setuptools)
- netCDF4 (select python3-netcdf4)

Click "Next" and finish the installation

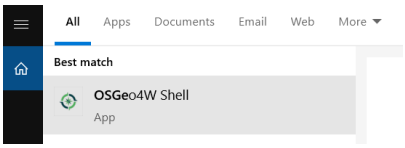
1.3.2 Setting environment variables

Steps

- Right click "This PC" in Windows 10 **OR** "My Computer" in windows 7, go to *Properties -> Advaced system settings -> Advanced* tab -> *Environment variables -> System variables*.
- Click new and add new system variables. Add the **Variable name** and **Variable value** as shown below.
- **GDAL_DATA** set to `C:\OSGeo4W64\share\epsg_csv`
- Edit the variable **Path** in the **System variables** to add the path `C:\OSGeo4W64\bin` to the end followed by a semicolon (;) in windows 7 **OR** add this path as a new line in the path variable in Windows 10.

Step3 - Install additional dependencies

- In the program menu search for "OSGeo4W Shell" or if you have selected "Create icon on Desktop" option in the previous step, it should be in the desktop. Now open "OSGeo4W Shell"



- In the OSGeo4W Shell type in the following commands to install packages - *openpyxl, joblib*

```
# Install following packages
# First enable python 3 by typing the following command and 'enter'
py3_env
pip3 install openpyxl joblib
pip3 install grass_session
```

Warning: In case the above installation give `fatal error` then please try the following commands.

```
python -m pip3 install openpyxl joblib
python -m pip3 install grass_session
```

1.3.3 Test installation

To test whether the PySEBAL will run, open OSGeo4W Shell, and type following commands.

```
# After each command click enter
# Any line starting with '#' is comment line
# First enable python 3 by typing the following command and 'enter'
py3_env
# Change drive
D:
# Change to the directory with SEBAL code
cd PySEBAL_dev\SEBAL
# open python
python
# import one of the PySEBAL Script
import pysebal_py3
# If there are no errors, the installation is successful
# To exit from python
exit()
```

1.4 Installation in Linux

The below steps are tested in Ubuntu 18.04 LTS, it should also work in other Linux distibutions, you may have to adapt some of the installation steps accordingly. This is also valid for installation in **Bash for Windows** app with Ubuntu inside windows 10.

Note: You can check the python version using the command `python --version` in a terminal

1.4.1 Installing dependencies

The dependencies packages are same as those in windows except for msys. We also install git to download and clone the PySEBAL_dev repository.

Open a Terminal and type in following commands to install required packages. You should have admin rights to install packages.

Warning: Please remove all the QGIS and GRASS packages you may have installed from other repositories before doing the update.

```
# After each command click enter
# Any line starting with '#' is comment line
```

(continues on next page)

(continued from previous page)

```
# Install git
sudo apt-get install git
# Add a PPA to install required GIS softwares
sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
# Install qgis and qgis-grass plugin
sudo apt-get install qgis qgis-plugin-grass
# Install GRASS GIS and required packages
sudo add-apt-repository ppa:ubuntugis/ppa
sudo add-apt-repository ppa:grass/grass-stable
sudo apt-get update
sudo apt-get install grass78
# Install openpyxl, netCDF4, joblib packages
# For python 3, use pip3 to install ....
pip install openpyxl netCDF4 joblib
```

For other Linux distributions there is detailed instruction to install qgis here and grass gis here.

1.4.2 Download source code

Open a terminal and type in following git command to download the PySEBAL_dev repository.

```
# After each command click enter
# Any line starting with '#' is comment line
# change to working directory,
# /mnt/d if you are accessing windows D: drive from linux. For example "bash for␣
↪windows" in windows 10
cd /mnt/d
# Clone the PySEBAL_dev repository
git clone https://github.com/spareeth/PySEBAL_dev.git
```

1.4.3 Testing installation

Open a terminal and type in following codes to test if the installation is successful.

```
# After each command click enter
# Any line starting with '#' is comment line
# change to the PySEBAL_dev directory, assuming that the repository is cloned in /mnt/
↪d
cd /mnt/d/PySEBAL_dev/SEBAL
# List the files inside this folder
ls
# Open Python
python
import pysebal_py3
# If there are no errors, the installation is successful
# To exit from python (ctrl-d)
exit()
```

1.5 Test run PySEBAL

Once PySEBAL is installed, we can run the PySEBAL code using the test data provided with the PySEBAL_dev library. The test data is located in the folder `PySEBAL_dev\test_data`. If you have installed PySEBAL in `D:` drive then it should be `D:\PySEBAL_dev\test_data`.

Assuming that PySEBAL_dev is in `D:` drive, Let us run the library with test data.

Open a OSGeo4W Shell and change the directory to `D:\PySEBAL_dev\SEBAL` and follow the commands given below.

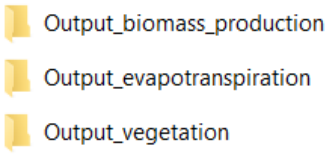
In Windows

```
# After each command click enter
# Any line starting with '#' is comment line
# First enable python 3 by typing the following command and 'enter'
py3_env
# change to the PySEBAL_dev\SEBAL directory
cd D:\PySEBAL_dev\SEBAL
# Run the PySEBAL script
python Run_py3.py
```

In Linux

```
# After each command click enter
# Any line starting with '#' is comment line
# change to the PySEBAL_dev\SEBAL directory
cd \mnt\d\PySEBAL_dev\SEBAL
# Run the PySEBAL script
python Run_py3.py
```

After the above commands, there will be a `output` folder inside `D:\PySEBAL\test_data` with the following structure.



Warning: If PySEBAL_dev is not in `D:` drive, adapt changes to the path in above commands accordingly. To change the path open the excel sheet `D:\PySEBAL_dev\docs\InputEXCEL_v3_3_7_WIN.xlsx` in case of Windows OR open `D:\PySEBAL_dev\docs\InputEXCEL_v3_3_7_LIN.xlsx` in case of Linux. You need to change the path in columns B, C & E in the sheet 1.

Note: Now go to the folder `D:\PySEBAL_dev\test_data\output\Output_evapotranspiration` and check the daily ETa map (*L8_ETact_24_30m_2014_03_10_069.tif*) in QGIS.

CHAPTER 2

Linux in Windows 10

This tutorial explain steps to install a Linux kernel 'Windows Subsystem for Linux (WSL)' - Ubuntu in your Windows 10 Computer.

WSL provides a Windows subsystem with Ubuntu (or other distros like SUSE) Linux runs atop it. It is not a virtual machine or an application like Cygwin. It is complete Linux system inside Windows 10. It allows you to run the same Bash shell that you find on Linux. This way you can run Linux commands inside Windows without the needing to install a virtual machine, or dual boot Linux and Windows. You install Linux inside Windows like a regular application. This is a good option if your main aim is to learn Linux/Unix commands. You can find more information here.

2.1 Requirements

This feature is available only in **Windows 10**.

2.2 Installation

2.2.1 Step 1: Enable "Windows Subsystem for Linux" feature

Go to start programs and type in "turn". Select "Turn Windows features on or off" as shown in figure below:

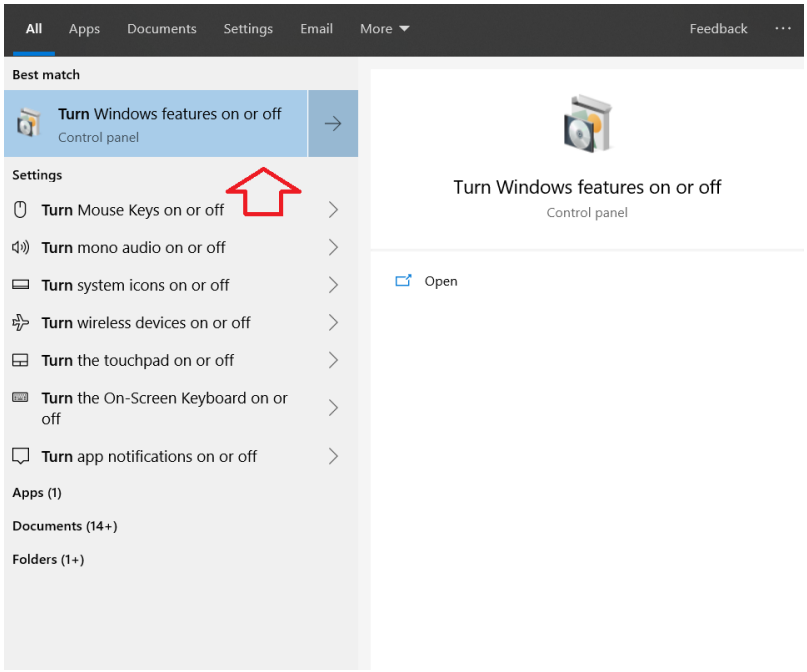
Enable (or check) the feature for **Windows Subsystem for Linux** and **Virtual Machine Platform** and then **restart** your computer to make sure you have both of them enabled.

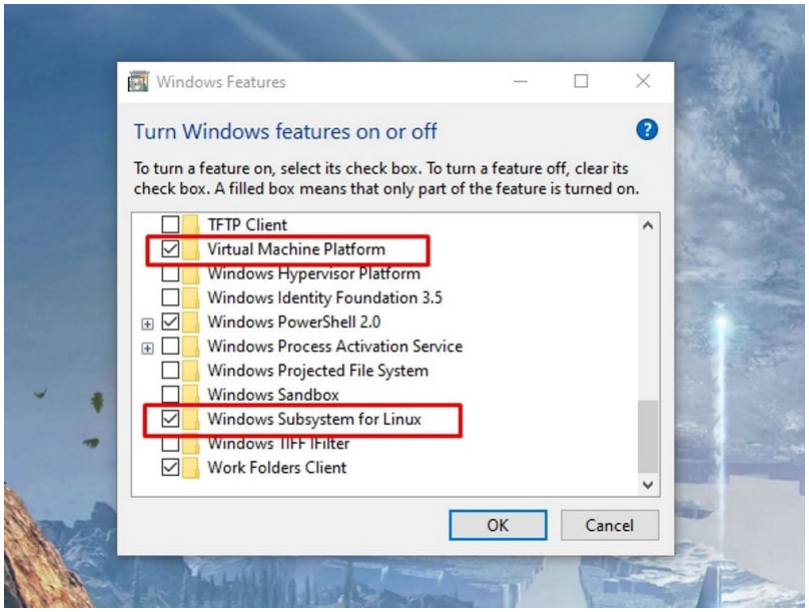
RESTART your Computer !

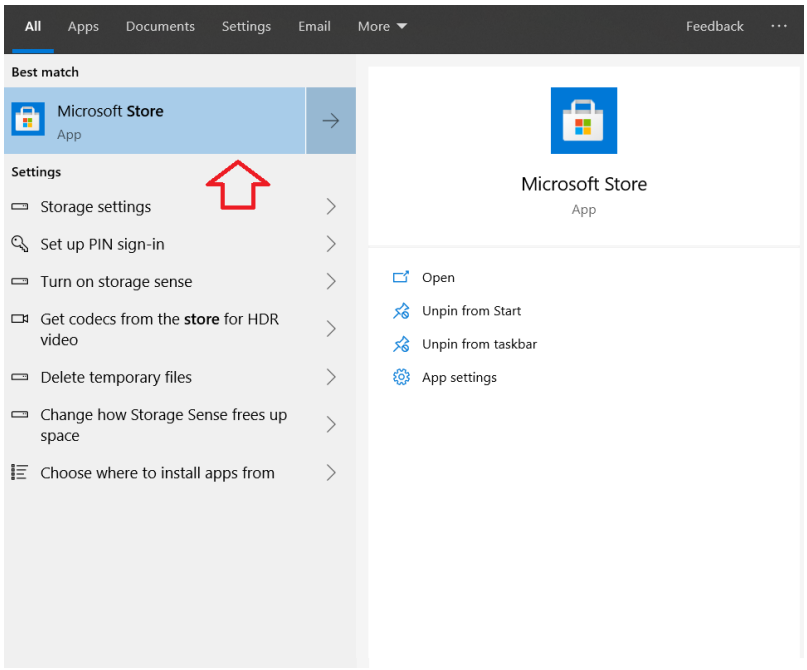
2.2.2 Step 2: Download a Linux system from the Windows store

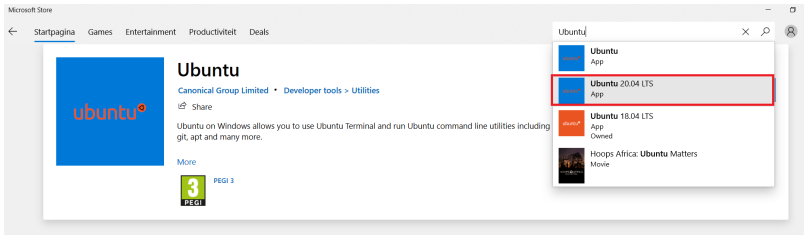
Once your system has restarted, go to the Windows Store and search for "Ubuntu".

Select **Ubuntu 20.04 LTS** and install it by clicking on "Get" button on top right. You need to be connected to internet for the computer to download the Ubuntu distro and install in your computer.


All
Apps
Documents
Settings
Email
More
Feedback
Best match
Turn Windows features on or off
Control panel
Settings
Turn Mouse Keys on or off
Turn mono audio on or off
Turn system icons on or off
Turn wireless devices on or off
Turn the touchpad on or off
Turn the On-Screen Keyboard on or off
Turn app notifications on or off
Apps (1)
Documents (14+)
Folders (1+)
Turn Windows features on or off
Control panel
Open


Windows Features
Turn Windows features on or off
To turn a feature on, select its check box. To turn a feature off, clear its check box. A filled box means that only part of the feature is turned on.
TFTP Client
Virtual Machine Platform
Windows Hypervisor Platform
Windows Identity Foundation 3.5
Windows PowerShell 2.0
Windows Process Activation Service
Windows Projected File System
Windows Sandbox
Windows Subsystem for Linux
Work Folders Client
OK
Cancel


All
Apps
Documents
Settings
Email
More
Feedback
Best match
Microsoft Store
App
Settings
Storage settings
Set up PIN sign-in
Turn on storage sense
Get codecs from the store for HDR video
Delete temporary files
Change how Storage Sense frees up space
Choose where to install apps from
Microsoft Store
App
Open
Unpin from Start
Unpin from taskbar
App settings


Microsoft Store
Startpagina
Games
Entertainment
Productiviteit
Deals
Ubuntu
Ubuntu
Canonical Group Limited • Developer tools > Utilities
Share
Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including git, apt and many more.
More
PEGI 3
Ubuntu
App
Ubuntu 20.04 LTS
App
Ubuntu 18.04 LTS
App
Owned
Hoops Africa: Ubuntu Matters
Movie

2.2.3 Step 3: Run Linux inside Windows 10

Once you have installed Linux, Let us run it inside the Windows.

Just search for Ubuntu in the Start programs. Click on Ubuntu which will open a command line terminal as shown below. You'll see that it runs like a normal Windows application. Only the first time, it will take some time to setup and it will also ask you to set up a username and a password.

> **Warning:** Password will not appear in the terminal (command line)

Now you are ready to use Linux inside Windows 10.

2.3 Additional Software

2.3.1 MobaXterm

MobaXterm is a X server which enables user interface features for programs installed in Linux (Ubuntu) within Windows 10 in this case.

Go to this link, download the home edition and install MobaXterm with default settings.

2.3.2 Spatial libraries in ubuntu

Now let us also install some spatial libraries required for data processing in Ubuntu.

Go to Start programs and open Ubuntu terminal.

```
# Run the following commands to install gdal, proj, grass etc.
sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
sudo apt-get install grass grass-gui grass-core grass-doc grass-dev
```

CHAPTER 3

PySEBAL data requirements

To run PySEBAL we need the following input data as shown below in the figure.

3.1 Satellite data

Currently PySEBAL support data from Landsat 4/5/7/8, MODIS, PROBA-V/VIIRS sensors/satellites. In this documentation currently PySEBAL using Landsat data as input is explained. But data from other satellites can be easily used by replacing the Landsat data,

3.1.1 Acquiring Landsat data

The main archive of Landsat satellite (all missions – TM/ETM/OLI) is earth explorer website: https://earthexplorer.usgs.gov/.

Note: First step is to create user login for this website so that you are able to download the data.

Let us now search and download data for Miandoab irrigation scheme in Iran for the time period April to September 2018.

Step1

In the "Search Criteria" tab type in your "address/place" of your interest, in this case "Miandoab" and click "enter" or search for Path/row - 168/034.

Click on the result (red box) and a location popup will appear over on the map.

Step2

Now enter the date range for which data is required in the same tab.

Step3

Select the datasets you want to search and set the additional criteria of those data with cloud cover less than 20%

Variable	Parameter	Unit	Description
		Satellite data	
Visible	R,G,B	-	Spectral reflectances from satellite sensors like Landsat, MODIS, Proba-V from the visible spectrum.
Near infrared	NIR	-	Spectral reflectances from satellite sensors like Landsat, MODIS, Proba-V from the NIR spectrum.
Short wave infrared	SWIR	-	Spectral reflectances from satellite sensors like Landsat, MODIS, Proba-V from the SWIR spectrum.
Thermal infrared	TIR	K	Thermal data from satellite sensors like Landsat, MODIS, Proba-V from the TIR spectrum.
		Meteorological data	
Downward shortwave radiation	SW_{down}	W/m^2	Total amount of shortwave radiation (both direct and diffuse) that reaches the Earth's surface.
Wind speed	W_S	m/s	Wind speed at 2m height.
Air temperature	T_{air}	°C	Air temperature at 2m height.
Pressure	P	Mb	Air pressure at 2m height.
Relative humidity	Rh	%	Amount of water vapour present in air expressed as a percentage of the amount needed for saturation at the same temperature.
		Topography	
Digital Elevation Model	DEM	M	Height of the land surface above mean sea level.
		Soil hydraulic properties	
Saturated water content	WC_{sat}	m^3/m^3	Saturated water content is the maximum amount of water a soil can store.
Residual water content	WC_{red}	m^3/m^3	Water content for which the gradient d(volumetric water content)/dh becomes zero.
Field capacity	WC_PF2	m^3/m^3	Field capacity is the amount of water content in the soil after excess water has drained away.
Wilting point	$WC_PF4.2$	m^3/m^3	Wilting point is defined as the minimum amount of water in the soil that the plant requires not to wilt.

Fig. 1: List of input data required for PySEBAL

Fig. 2: Setting the search criteria

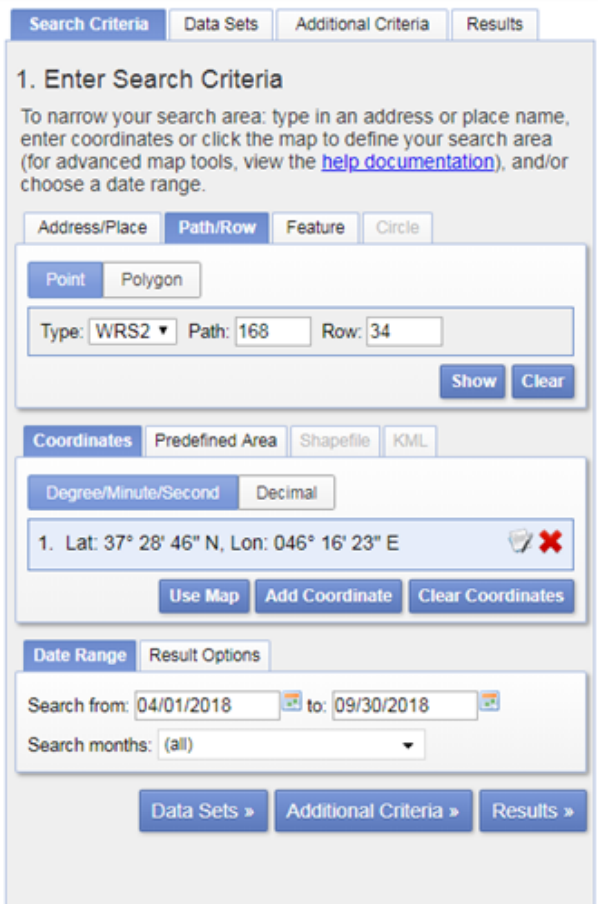


Fig. 3: Setting the date range

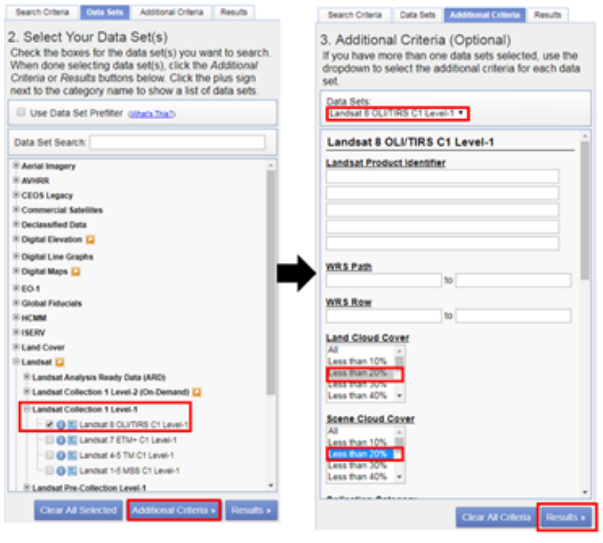


Fig. 4: Setting additional criteria and listing results

Here we are selecting only Landsat 8 data. For previous years you can also try with Landsat 7 ETM, and Landsat 4/5 TM. Finally click the "Results" to see list of all available data with our conditions met for the study area in the given period of time.

Step4

Check the listed scenes using browse images, select and download the 168/034 scene

Fig. 5: List of available landsat 8 data for the given period of time

The image icon under each result can be used to see the preview of that landsat scene (see Figure above). The download icon can be used to download that single scene with out ordering. While the Bulk download icon can be used to get a single link to download multiple products at a time. More details on bulk download can be found here - https://www.usgs.gov/media/videos/eros-earthexplorer-how-do-a-bulk-download.

For example, image dated 22 June 2018 looks really good, click on the download icon to get the data. You have to login in order to download the data.

From the list of options download the "Level-1 GeoTIFF Data Product" to get all the spectral bands and metadata of the scene.

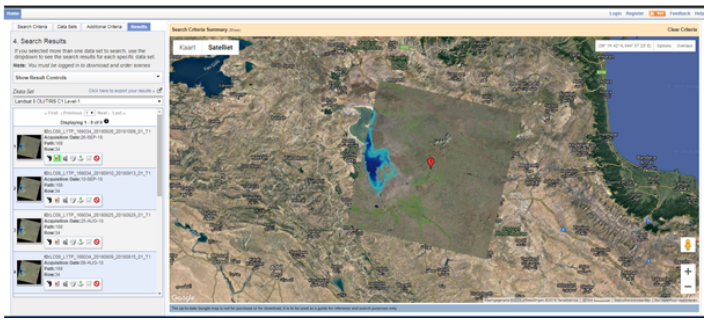


Fig. 6: Showing the preview image of a landsat scene

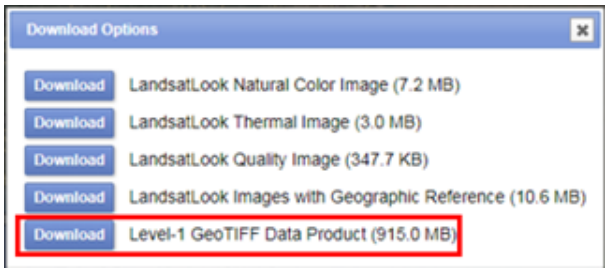


Fig. 7: Landsat download options, select the one highlighted

3.1.2 Bulk download Landsat data using command line

This section explains how you can download big amount of Landsat data from Google cloud bucket using command line.

Steps

- Install gsutil library - https://pypi.org/project/gsutil/
- If you want to list all the Landsat 8 data over Miandoab covering tile 168/034 in June 2018 use the following command:

```
gsutil ls -d gs://gcp-public-data-landsat/LC08/01/168/034/LC08_L1TP_168034_201806*T1
```

- To download them, use the following command:

```
# '.' means it will download to present directory
gsutil -m cp -r gs://gcp-public-data-landsat/LC08/01/168/034/LC08_L1TP_168034_
↪201806*T1 .
```

This link has more details on this approach.

3.1.3 Naming convention of Landsat data

3.2 Meteo data

Meteo data is either obtained from field stations or from global models like GLDAS, ERA5, MERRA2 etc. Here we will use instantaneous and daily average computed from GLDAS data. You can download 3 hourly GLDAS data from this web link: https://hydro1.gesdisc.eosdis.nasa.gov/data/GLDAS/GLDAS_NOAH025_3H.2.1/

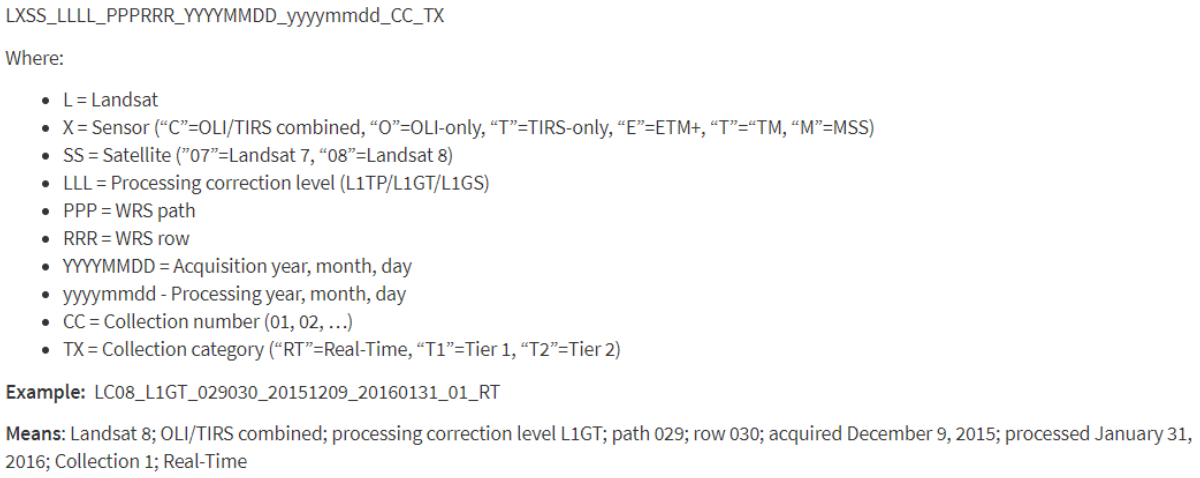


Fig. 8: Landsat data naming convention

Once downloaded the data, we use GDAL and GRASS GIS to compute the required meteo parameters for PySEBAL.

The GLDAS data is provided in netCDF4 format with number of measured parameters as subdatasets. Let us assume that you have downloaded the GLDAS data for the 6 June 2018 00:00 hours.

The file name will be `GLDAS_NOAH025_3H.A20180606.0000.021.nc4`.

Now let us do all the processing in GDAL library and GRASS GIS already installed in your system.

Open **OSGeo4W Shell** Type `grass78 --gui` and enter It will open the following interface

For an introduction to GRASS GIS see this presentation.

Before we proceed with GRASS GIS we will set the linux environment in the OSGeo4W Shell. Please download this file and save in `echo $HOME` folder. Please change line no: 22 in this file `.bashrc` only the `/c/OSGeo4W64/apps/grass/grass78/scripts` to the corresponding path in your computer.

```
# Start the Linux bash
bash
```

Warning: In the export path above, adapt your path accordingly. If your OSGeo4W installation is elsewhere, make changes accordingly.

Now in the command line type in following commands to extract required variables from GLDAS

```
# To see the metadata run the following command
gdalinfo GLDAS_NOAH025_3H.A20180606.0000.021.nc4
# To import specific humidity
gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.0000.021.nc4":Qair_f_inst GLDAS_
↪NOAH025_3H_20180606_0000_Qair.tif
# To import Pressure in Pa
gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.0000.021.nc4":Psurf_f_inst GLDAS_
↪NOAH025_3H_20180606_0000_Psurf.tif
# To import air temperature
gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.0000.021.nc4":Tair_f_inst GLDAS_
↪NOAH025_3H_20180606_0000_Tair.tif
# To import Wind speed
```

(continues on next page)

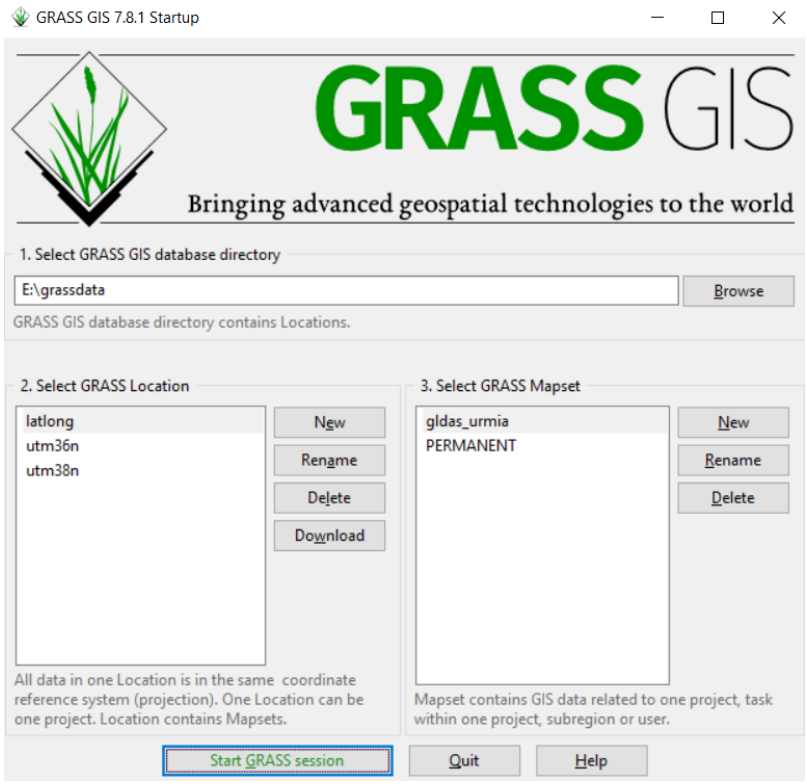


Fig. 9: GRASS GIS start window, set DB, Location and Mapset here.

(continued from previous page)

```
gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.0000.021.nc4":Wind_f_inst GLDAS_
↪NOAH025_3H_20180606_0000_Wind.tif
# To import Short wave downward radiation
gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.0000.021.nc4":SWdown_f_tavg GLDAS_
↪NOAH025_3H_20180606_0000_SWdown.tif
```

** How to do the above set of commands using a `for` loop **

```
# To convert the Wind speed parameter for a day every three hours data, means 8 tif␣
↪files for wind speed a day
for i in "00" "03" "06" "09" "12" "15" "18" "21"; do
    gdal_translate NETCDF:"GLDAS_NOAH025_3H.A20180606.${i}00.021.nc4":Qair_f_inst␣
↪GLDAS_NOAH025_3H_20180606_${i}00_Qair.tif
done
# Now repeat it for all the 6 parameters required for PySEBAL for both dates - 06␣
↪June 2019 and 22 June 2019
```

Now we have to import these `tif` files into GRASS GIS Following commands will import the files into GRASS GIS

```
# To import all the above tif files
r.import.py in=GLDAS_NOAH025_3H_20180606_0000_Qair.tif out=GLDAS_NOAH025_3H_20180606_
↪0000_Qair -o --o
r.import.py in=GLDAS_NOAH025_3H_20180606_0000_Psurf.tif out=GLDAS_NOAH025_3H_
↪20180606_0000_Psurf -o --o
r.import.py in=GLDAS_NOAH025_3H_20180606_0000_Tair.tif out=GLDAS_NOAH025_3H_20180606_
↪0000_Tair -o --o
```

(continues on next page)

(continued from previous page)

```
r.import.py in=GLDAS_NOAH025_3H_20180606_0000_Wind.tif out=GLDAS_NOAH025_3H_20180606_0000_Wind -o --o
r.import.py in=GLDAS_NOAH025_3H_20180606_0000_SWdown.tif out=GLDAS_NOAH025_3H_20180606_0000_SWdown -o --o
```

Note: Try to do the above commands and the following commands using `for` loop

Let us set the Computational region in GRASS GIS so that rest of all the analysis compute only in our study area

```
# To set the computational region
g.region res=0.25 -a
g.region -p
```

Warning: Always start your GRASS GIS work with checking the `g.region -p` to make sure about the computational region and resolution.

Now we have to do three major Steps * Convert airtemperature in kelvin to Deg C. * Convert Pressure in Pa to Milli bar (Mb) * Convert Specific humidity to relative humidity following the description here

Run the following commands to do the conversions:

```
## Air temperature
r.mapcalc "GLDAS_NOAH025_3H_20180606_0000_Tair_deg = GLDAS_NOAH025_3H_20180606_0000_Tair - 273.15" --o
    ## Pressure convert from pa to mb
r.mapcalc "GLDAS_NOAH025_3H_20180606_0000_Psurf_mb = GLDAS_NOAH025_3H_20180606_0000_Psurf / 100" --o
    ## Humidity according to the url: https://earthscience.stackexchange.com/questions/2360/how-do-i-convert-specific-humidity-to-relative-humidity
# Saturation vapour pressure
r.mapcalc "es = 6.112 * exp((17.67 * GLDAS_NOAH025_3H_20180606_0000_Tair_deg) / (GLDAS_NOAH025_3H_20180606_0000_Tair_deg + 243.5))" --o
# vapour pressure
r.mapcalc "e = (GLDAS_NOAH025_3H_20180606_0000_Qair * GLDAS_NOAH025_3H_20180606_0000_Psurf_mb) / (0.378 * GLDAS_NOAH025_3H_20180606_0000_Qair + 0.622)" --o
# Calculate Relative humidity
r.mapcalc "GLDAS_NOAH025_3H_20180606_0000_Rh1 = (e / es) * 100" --o
# Remove outliers
r.mapcalc "GLDAS_NOAH025_3H_20180606_0000_Rh = float(if(GLDAS_NOAH025_3H_20180606_0000_Rh1 > 100, 100, if(GLDAS_NOAH025_3H_20180606_0000_Rh1 < 0, 0, GLDAS_NOAH025_3H_20180606_0000_Rh1)))" --o
```

Note: How to do above set of commands in a single run using `for` loop ??

Repeat the above steps for other NC files as well, GLDAS_NOAH025_3H.A20180606.0300.021.nc4, GLDAS_NOAH025_3H.A20180606.0600.021.nc4, GLDAS_NOAH025_3H.A20180606.0900.021.nc4, GLDAS_NOAH025_3H.A20180606.1200.021.nc4, GLDAS_NOAH025_3H.A20180606.1500.021.nc4, GLDAS_NOAH025_3H.A20180606.1800.021.nc4, GLDAS_NOAH025_3H.A20180606.2100.021.nc4

Using single commands **OR** combine jobs using `for` loop

Now let us create instantaneous and daily averages: For the data in 20180606 follow the commands below in GRASS

GIS, For instantaneous we are going to take the data at 0900 hour as Landsat acquisition time is around 8:30 (Both Landsat and GLDAS times are in GMT). The time varies according to your study area.

```
## Air temperature instantaneous
r.mapcalc "GLDAS_NOAH025_3H_20180606_Tair_inst = GLDAS_NOAH025_3H_20180606_0900_Tair_
↪deg"
## Shortwave radiation instantaneous
r.mapcalc "GLDAS_NOAH025_3H_20180606_SWdown_inst = GLDAS_NOAH025_3H_20180606_0900_
↪SWdown"
## Wind speed instantaneous
r.mapcalc "GLDAS_NOAH025_3H_20180606_Wind_inst = GLDAS_NOAH025_3H_20180606_0900_Wind"
## Relative humidity instantaneous
r.mapcalc "GLDAS_NOAH025_3H_20180606_Rh_inst = GLDAS_NOAH025_3H_20180606_0900_Rh"
```

Note: How to do above set of commands in a single run using `for` loop ??

Next calculate the daily averages

```
## Air temperature daily average
MAPS1=`g.list rast pattern=GLDAS_NOAH025_3H_20180606_*_Tair_deg$ sep=, map=.|cat`
r.series input=${MAPS1} output=GLDAS_NOAH025_3H_20180606_Tair_24 method=average
## Short wave radiation daily average
MAPS2=`g.list rast pattern=GLDAS_NOAH025_3H_20180606_*_SWdown$ sep=, map=.|cat`
r.series input=${MAPS2} output=GLDAS_NOAH025_3H_20180606_SWdown_24 method=average
## Wind daily average
MAPS3=`g.list rast pattern=GLDAS_NOAH025_3H_20180606_*_Wind$ sep=, map=.|cat`
r.series input=${MAPS3} output=GLDAS_NOAH025_3H_20180606_Wind_24 method=average
## Relative humidity daily average
MAPS4=`g.list rast pattern=GLDAS_NOAH025_3H_20180606_*_Rh$ sep=, map=.|cat`
r.series input=${MAPS4} output=GLDAS_NOAH025_3H_20180606_Rh_24 method=average
```

Note: How to do above set of commands in a single run using `for` loop ??

Now let us **resample** the instantaneous and daily averaged to avoid pixel effects and **export** the prepared raster maps to tif files for PySEBAL to read and process Landsat data.

```
## Set the region with require resolution
g.region vect=study_area_big res=0.0625 -a
## change directory to output folder
cd /to/the/folder/you/want/to/store/meteo/data
## For loop to resample all the instantaneous maps
for i in `g.list rast pattern=*inst$ map=.`; do
    r.resamp.bspline in=${i} out=${i}_interp method=bicubic --o
    r.out.gdal in=${i}_interp out=${i}_interp.tif --o
done
## For loop to resample all the daily averages maps
for i in `g.list rast pattern=*24$ map=.`; do
    r.resamp.bspline in=${i} out=${i}_interp method=bicubic --o
    r.out.gdal in=${i}_interp out=${i}_interp.tif --o
done
```

Some usful GRASS GIS documentation and links:

- Module documentation
- GRASS GIS Wiki

- GRASS intro workshop held at NCSU
- GRASS GIS course in Jena 2018
- GRASS GIS course IRSAE 2018
- GRASS GIS course in Argentina 2018

CHAPTER 4

PySEBAL data preparation and execution

4.1 Input data preparation

Let us now arrange all the data we prepared to Run PySEBAL and prepare a input excel sheet.

Assuming that you have following folder structure (recommended) for input data:

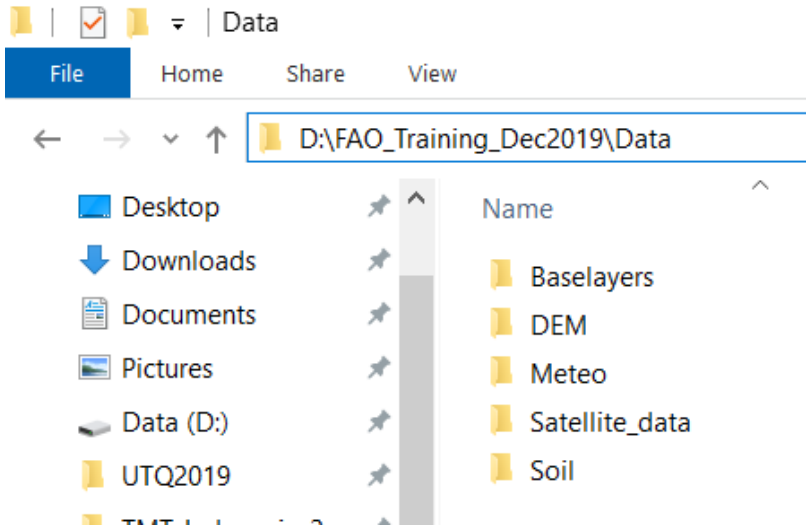


Fig. 1: Folder structure of input data

Open the excel sheet provided to you - `D:\PySEBAL_dev\docs\InputEXCEL_v3_3_7_WIN.xlsx` and make necessary changes to the excel sheet as listed below.

Sheet 1: General Input

- **InputMap** - set the path to your satellite data
- **OutputMap** - Where you want to save PySEBAL outputs (if not existing, PySEBAL will create the folder)

- **Image_type** - satellite type
- **NameDEM** - set the path to DEM file (Note that you need to provide the file name with extension(.tif))

Sheet 2: Meteo_Input

- **Temp_inst** - set the path to instantaneous air temperature (including file name and extension)
- **Temp_24** - set the path to daily average air temperature (including file name and extension)
- **RH_inst** - set the path to instantaneous Relative humidity (including file name and extension)
- **RH_24** - set the path to daily average Relative humidity (including file name and extension)
- **Wind_inst** - set the path to instantaneous wind speed (including file name and extension)
- **Wind_24** - set the path to daily average wind speed (including file name and extension)
- **Rs_inst** - set the path to instantaneous downward shortwave radiation (including file name and extension)
- **Rs_24** - set the path to daily downward shortwave radiation (including file name and extension)

Sheet 3: Soil_Input

- **Saturated soil moisture content** - set the path with filename and extension
- **Saturated soil moisture content subsoil** - set the path with filename and extension
- **Residual soil moisture content** - set the path with filename and extension
- **Residual soil moisture content subsoil** - set the path with filename and extension
- **Field_Capacity** - set the path with filename and extension
- **Wilting point** - set the path with filename and extension

Sheet 4: Landsat_Input

- **Name Landsat Image** - Name of the landsat image bands (for example without `_B1.TIF`)
- **Landsat Number** - 4/5/7/8 depending which landsat
- **Bands Thermal** - 1/2, In case of Landsat 8, it is 2
- **tscold_min** - Min percentile to compute minimum threshold for cold pixel from temperature layer (Default is 5)
- **tscold_max** - Max percentile to compute maximum threshold for cold pixel from temperature layer (Default is 10)
- **ndvihot_low** - Min percentile to compute minimum threshold from NDVI layer (Default is 2)
- **ndvihot_high** - Max percentile to compute maximum threshold from NDVI layer (Default is 5)
- **temp_lapse_rate** - Temperature lapse rate for correction of surface temperature

---

Note: Number of Rows in the input excel sheet is equal to the number of landsat images you want to process. If you have 10 images, the row numbers are from 2 to 11.

Once the input excel sheet is ready, open the Run SEBAL python file (`Run_py3.py`) which is in `D:\PySEBAL_dev\SEBAL` folder.

Open the file `Run_py3.py` in Notepadd++.

Edits in Run_py3 file

We need to make following changes in this file:

- **Line 14** - Set the path to prepared excel sheet

- **Line 15/16** -Set start and end row numbers for running all the landsat images in one go.

4.2 Run PySEBAL

Once you made the changes save and close the file `Run_py3`

Now open **new OSGeo4W Shell** and cd to `PySEBAL_dev\SEBAL` folder and run the following command.

```
python Run_py3.py
```

Note: In the **Sheet 4: Landsat_Input** of the input excel sheet we have to set NDVI and Temperature min and max percentile thresholds for cold and hot pixels. As a rule of thumb, we can use 5th and 10th percentile from corrected surface temperature as low and high cold pixel thresholds. We use 2nd and 5th percentile from NDVI as low and high hot pixel thresholds. If the ETa results are not desirable with the default values, you may want to try different combinations. Also for a specific region, one set of values seems to work.

4.3 Output data structure

Once the PySEBAL run successfully, you will find following structure in the output folder (one set in the excel sheet !)

4.4 Details of the output data

Once the PySEBAL run successfully, you will find following data in the output folder (one set in the excel sheet !)

log.txt - All the constants and derived thresholds are stored here

Folder 1: Output_biomass_production

- **L8_Biomass_production_30m_2014_03_10_069.tif** - Biomass production (Kg/ha)
- **L8_Biomass_wp_30m_2014_03_10_069.tif** - Biomass Water Productivity WPb (Kg/m3)
- **L8_Biomass_deficit_30m_2014_03_10_069.tif** - Deficit Biomass production (Kg/ha)

Folder 2: Output_evapotranspiration

- **L8_Advection_Factor_30m_2014_03_10_069.tif** - Advection factor (unitless)
- **L8_EFinst_30m_2014_03_10_069.tif** - instantaneous Evaporative Fraction (unitless)
- **L8_ET_24_deficit_30m_2014_03_10_069.tif** - 24 hours ET deficit (mm/day)
- **L8_ETact_24_30m_2014_03_10_069.tif** - 24 hours Actual EvapoTranspiration (mm/day)
- **L8_ETpot_24_30m_2014_03_10_069.tif** - 24 hours Potential EvapoTranspiration (mm/day)
- **L8_ETref_24_30m_2014_03_10_069.tif** - 24 hours Reference EvapoTranspiration (mm/day)
- **L8_Eact_24_30m_2014_03_10_069.tif** - 24 hours Actual Evaporation (mm/day)
- **L8_T_24_deficit_30m_2014_03_10_069.tif** - 24 hours Deficit Transpiration (mm/day)
- **L8_Tact_24_30m_2014_03_10_069.tif** - 24 hours Actual Transpiration (mm/day)
- **L8_Tpot_24_30m_2014_03_10_069.tif** - 24 hours Potential Transpiration (mm/day)

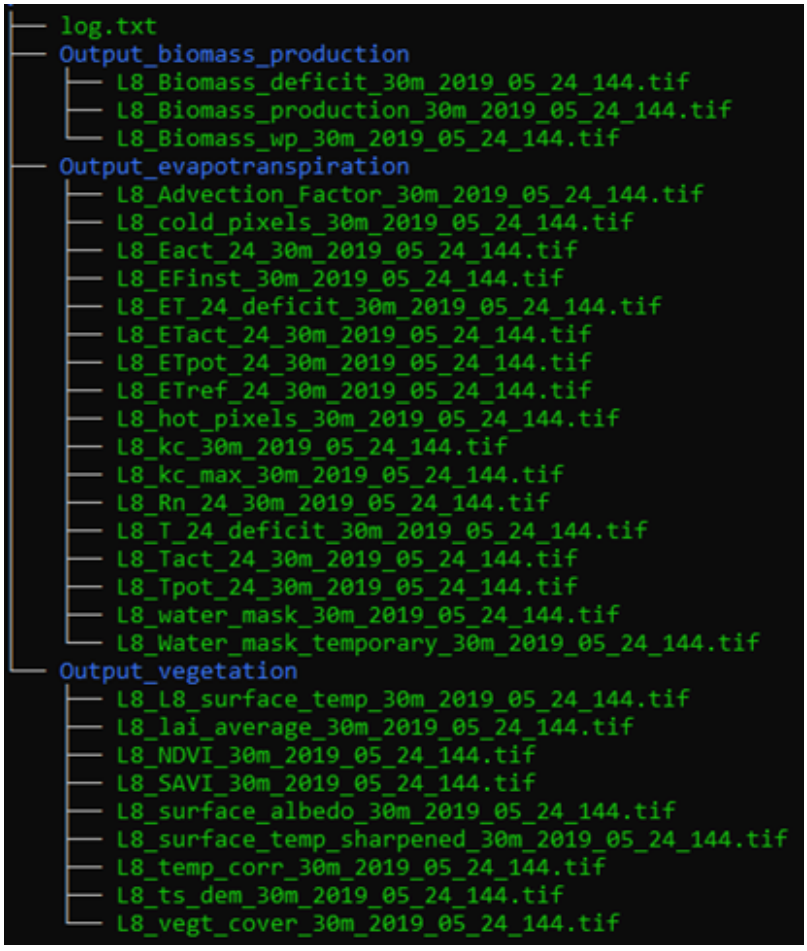


Fig. 2: Folder structure of output data

- **L8_cold_pixels_30m_2014_03_10_069.tif** - Detected cold pixels (unitless)
- **L8_hot_pixels_30m_2014_03_10_069.tif** - Detected hot pixels (unitless)
- **L8_kc_30m_2014_03_10_069.tif** - Crop coefficient Kc (unitless)
- **L8_kc_max_30m_2014_03_10_069.tif** - Max Crop coefficient Kc (unitless)
- **L8_water_mask_30m_2014_03_10_069.tif** - Water mask (unitless)

Folder 3: Output_vegetation

- **L8_L8_surface_temp_30m_2014_03_10_069.tif** - TOA temperature (Kelvin)
- **L8_NDVI_30m_2014_03_10_069.tif** - Normalized Diffrence Vegetation Index (unitless)
- **L8_SAVI_30m_2014_03_10_069.tif** - Soil Adjusted Vegetation Index (unitless)
- **L8_lai_average_30m_2014_03_10_069.tif** - Leaf Area Index (unitless)
- **L8_surface_albedo_30m_2014_03_10_069.tif** - Surface albedo (unitless)
- **L8_surface_temp_sharpened_30m_2014_03_10_069.tif** - Sharpened Temperature using NDVI (Kelvin)
- **L8_temp_corr_30m_2014_03_10_069.tif** - Surface Temperature (Kelvin)
- **L8_ts_dem_30m_2014_03_10_069.tif** - DEM corrected Temperature (Kelvin)
- **L8_vegt_cover_30m_2014_03_10_069.tif** - vegetation cover (unitless)

CHAPTER 5

Aggregating to monthly and gapfilling

The aggregation and gapfill script available with PySEBAL will perform the monthly aggregation, filtering, temporal and spatial interpolation on outputs of PySEBAL runs. The below video explains how to run gapfill script on a set of output files from PySEBAL run.