Introduction to Sentinel-2 time-series data as a use-case example using MAJA cloud mask.

The work is structured in the following way: firstly, brief theoretical introduction is provided followed by aim and goals of the work, secondly, data sources are provided; thirdly, the sequence of actions to configure MAJA using Linux Subsystem is described; lastly, we run MAJA on the test data to obtain our first cloud mask. Some necessary theory related to data formats also provided within the steps.

Introduction.

A wide range of contemporary technologies provide plenty of possibilities for data collection and analysis. Remote sensing is one of the most advanced approaches nowadays that is, for example, widely used in agriculture. In this case, satellite images of the Earth provide information about agricultural parcels and minimize the necessity of ground observations. Since the amount of data is quite big, unsupervised classification is used to detect anomalies. There are, however, some obstacles that affect the classification results, quality and availability of the satellite data. The issue we are going to focus on within this work are the cloud cover. Clouds can be considered as one of the biggest issues that processing methods have to deal with, since cameras cannot see through them and classification methods, as a rule, have wrong results due to cloud cover. As a result, cloudy areas should be taken off from the result in order to avoid mistakes in calculations and misclassifications.

In this work we are going to utilize Sentinel-2 L1C time series data in MAJA (processor for cloud detection and atmospheric correction) algorithm to obtain a cloud mask. The main difference from previous tasks is data type: instead of using one image of the area, MAJA uses spatiotemporal data. To understand the structure of the data, two core features can be highlighted – spatial and temporal. Spatial or geospatial data can be seen as a set of numerical values in a geographic coordinate system (Ansari, 2020). To put it even more simple, physical objects (e.g. buildings, roads, waterbodies) are described as sets of values in respect to a starting point. The second feature of the data is temporal, which means that the spatial data is collected multiple times over a certain period of time, so it includes temporal information as well (Figure 1) (Ansari, 2020). In the end, the result can be seen as a highly dimensional and dynamic time-series dataset which has spatial information about the place of interest at different points in time (Aghabozorgi, 2015; Ansari, 2020).

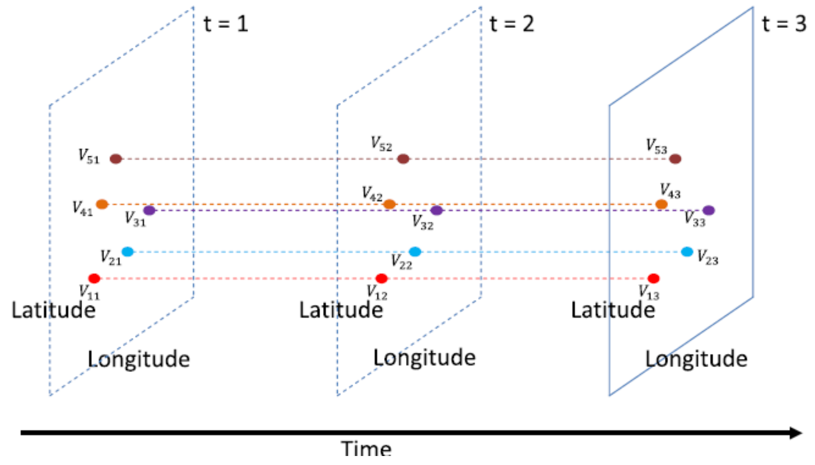


Figure 1. Geo-referenced time-series dataset
Source: Adapted from Ansari, 2020

Aim and goals of the work.

Considering the theoretical introduction, the aim of the work is to obtain cloud masks from Sentinel-2 time series data using MAJA.

The goals of the work are:

a) to introduce spatiotemporal data;
b) to introduce MAJA processor;
c) to practice obtaining and processing time-series data in MAJA.

Data sources and additional information.

1) CNES MAJA - https://logiciels.cnes.fr/en/content/maja
2) CNES MAJA Source code - https://github.com/CNES/MAJA
3) CREODIAS finder - https://finder.creodias.eu/
4) Copernicus DEM - https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1?tab=download
5) WSL2 - https://docs.microsoft.com/en-us/windows/wsl/install-win10#manual-installation-steps
6) Conda Installer - https://docs.conda.io/en/latest/miniconda.html#linux-installers
7) THEIA's L2A product format - https://labo.obs-mip.fr/multitemp/sentinel-2/theias-sentinel-2-l2a-product-format/#English

I. Windows preparation.

In this part you are going to get your computer ready for further work. This part is intended for Windows 10 users and in case if you are using any Linux OS, you can skip this chapter.

1) To start, you need to make sure that your system supports WSL2 (Windows Subsystem for Linux) technology. For that navigate to ***Windows settings -> About*** and check your version and build: for x64 systems - Version 1903 or higher, with Build 18362 or higher. Builds lower than 18362 do not support WSL 2.

2) If the previous requirement not met, update your system ***Windows settings -> Update & Security.*** If the previous requirement met, then you are going to manually install the feature support.

3) Enable the Windows Subsystem for Linux: for that navigate to ***Search -> PowerShell -> Open as Administrator***. PowerShell example can be seen on Figure 2. Within PowerShell you need to execute the following command and keep the shell opened:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

4) Enable Virtual Machine feature: in the same PowerShell window you need to execute the following command and keep the shell opened:

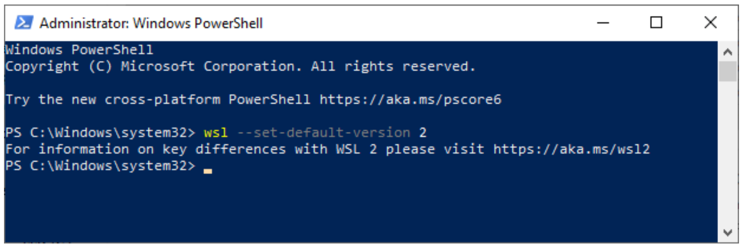
```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

For clarification, these steps just activate the system support for WSL2 which is turned off by default.

5) Next, you need to install the Linux Kernel Package that can be found on the ***WSL2 link, Data subsection, Step 4.*** You are looking for *WSL2 Linux kernel update package for x64 machines* which should be installed after downloading.

6) When the installation is done, set WSL 2 as your default version and close the shell:

```
wsl --set-default-version 2
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> wsl --set-default-version 2
For information on key differences with WSL 2 please visit https://aka.ms/wsl2
PS C:\Windows\system32>
```

Figure 2. PowerShell command example

7) Next, we need to download Ubuntu Linux from Microsoft Store, for that navigate to ***Search -> Microsoft Store -> Ubuntu -> Select first option.***

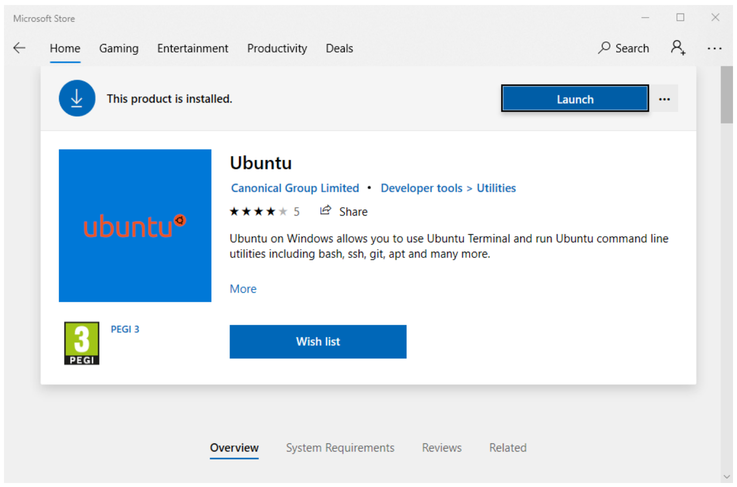

Microsoft Store
Home
Gaming
Entertainment
Productivity
Deals
Search
This product is installed.
Launch
Ubuntu
Canonical Group Limited • Developer tools > Utilities
★★★★★ 5
Share
Ubuntu on Windows allows you to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.
More
PEGI 3
Wish list
Overview
System Requirements
Reviews
Related

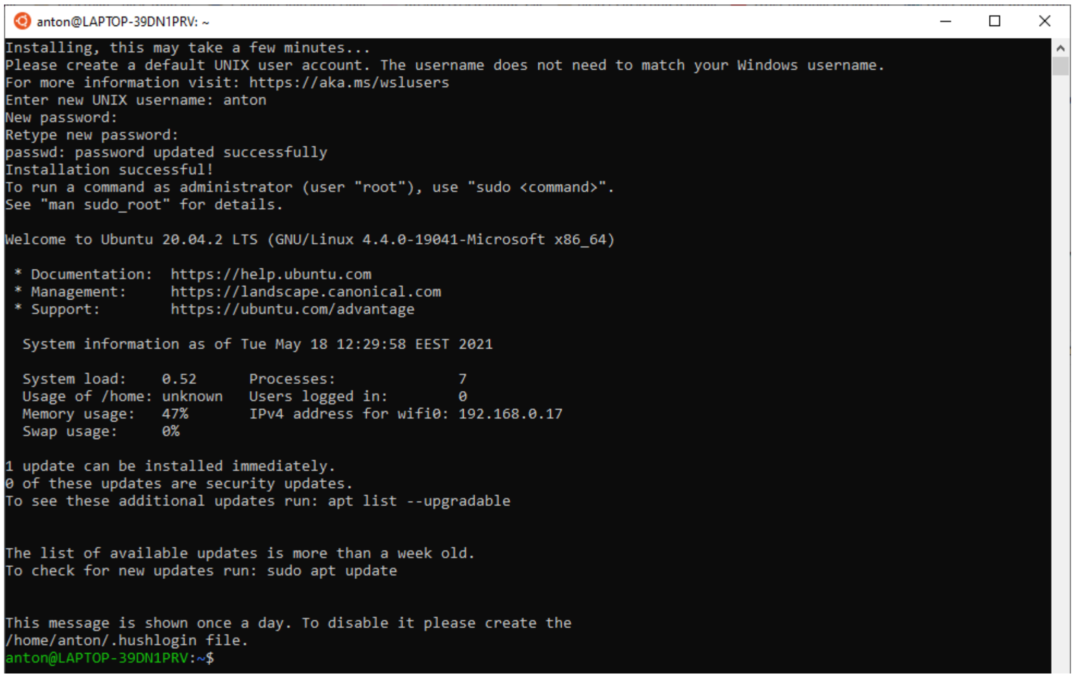
Figure 3. Ubuntu tool in MS Store

8) As soon as installation is done, you can ***start it by pressing an Ubuntu icon in start menu*** and go to the next chapter.

II. Ubuntu Linux configuration.

Here you are going to create an environment in your Ubuntu to make it start MAJA.

1) For windows users that just opened Ubuntu, it is necessary to ***create a user account*** with password. Pick some easy and short usernames (~5 symbols) and short passwords (note that password won't be displayed) this time around (Figure 4), since windows has some limitations on path length that can make it impossible for you to use subsystem anymore. Linux users are free to choose any combinations.



```
anton@LAPTOP-39DN1PRV: ~
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: anton
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue May 18 12:29:58 EEST 2021

  System load:    0.52      Processes:             7
  Usage of /home: unknown   Users logged in:       0
  Memory usage:   47%       IPv4 address for wifi0: 192.168.0.17
  Swap usage:     0%

1 update can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


This message is shown once a day. To disable it please create the
/home/anton/.hushlogin file.
anton@LAPTOP-39DN1PRV:~$
```

Figure 4. Ubuntu user creation

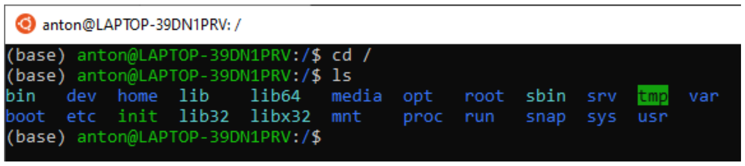
2) To create environment for MAJA you are going to use Miniconda. To obtain the installer, ***follow the miniconda link*** provided in the data subsection ***and choose a "Python 3.8 - Miniconda3 Linux 64-bit"*** option.

For windows users it is easier to put necessary files to Linux using default windows explorer. The root directory for Linux is by default located at the following path you can navigate to:

```
C:\Users\<YourWindowsUserName>\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\
LocalState\rootfs
```

Note that you are supposed to change <YourWindowsUserName>, in my case it is \anton\

Within rootfs folder you can see the same folder that you can get by typing ***cd /*** and ***ls*** in the Ubuntu terminal:



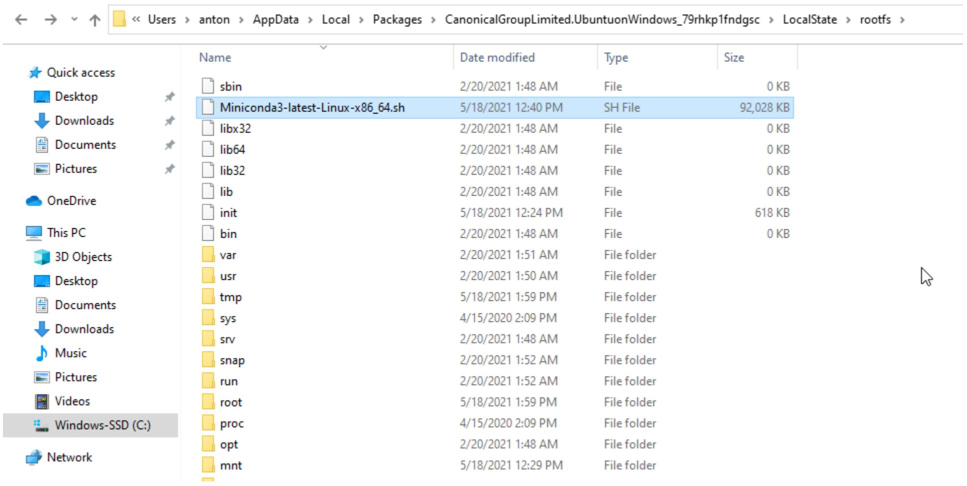
```
anton@LAPTOP-39DN1PRV: /
(base) anton@LAPTOP-39DN1PRV:/$ cd /
(base) anton@LAPTOP-39DN1PRV:/$ ls
bin   dev  home  lib    lib64   media  opt   root  sbin  srv  tmp  var
boot  etc  init  lib32  libx32  mnt    proc  run   snap  sys  usr
(base) anton@LAPTOP-39DN1PRV:/$
```

Figure 5. content of "/" folder

Hint!
ls – prints content of the current directory
cd /path/to/target – navigates to target folder
cd .. – allows to navigate to parent folder

To keep everything simple we are going to work in / directory.

3) Next step is to put downloaded Miniconda installer into "/". For that, ***copy your downloaded .sh file and paste it into rootfs folder*** in windows explorer. It should look like this:



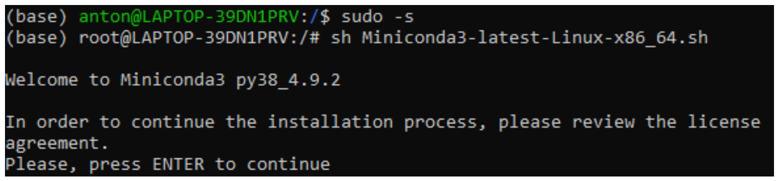
Name	Date modified	Type	Size
sbin	2/20/2021 1:48 AM	File	0 KB
Miniconda3-latest-Linux-x86_64.sh	5/18/2021 12:40 PM	SH File	92,028 KB
libx32	2/20/2021 1:48 AM	File	0 KB
lib64	2/20/2021 1:48 AM	File	0 KB
lib32	2/20/2021 1:48 AM	File	0 KB
lib	2/20/2021 1:48 AM	File	0 KB
init	5/18/2021 12:24 PM	File	618 KB
bin	2/20/2021 1:48 AM	File	0 KB
var	2/20/2021 1:51 AM	File folder	
usr	2/20/2021 1:50 AM	File folder	
tmp	5/18/2021 1:59 PM	File folder	
sys	4/15/2020 2:09 PM	File folder	
srv	2/20/2021 1:48 AM	File folder	
snap	2/20/2021 1:52 AM	File folder	
run	2/20/2021 1:52 AM	File folder	
root	5/18/2021 1:59 PM	File folder	
proc	4/15/2020 2:09 PM	File folder	
opt	2/20/2021 1:48 AM	File folder	
mnt	5/18/2021 12:29 PM	File folder	

Figure 6. Miniconda installer in "/" folder

4) Now if to run ***ls*** in the Ubuntu shell, you will be able to see Miniconda installer among the folders. In order to avoid problems with permissions we switch to root user. For that, run ***sudo -s, type your password and press enter.*** After that you will see that you are not who you are, but root (Figure 7). It gives you all access to all files within Linux subsystem and increases chances that you can break something down, so be careful.

5) Under root user you need to install miniconda, for that run ***sh Miniconda3-latest-Linux-x86_64.sh.*** After that it asks you if you want to continue, ***press enter, skip through the text using space, and agree on conda initialization*** in the end. After initialization, run ***exec bash*** that will restart your shell

configuration file and if everything good, then you have (base) written before root user as it is displayed on figure 7:



```
(base) anton@LAPTOP-39DN1PRV:/$ sudo -s
(base) root@LAPTOP-39DN1PRV:/# sh Miniconda3-latest-Linux-x86_64.sh

Welcome to Miniconda3 py38_4.9.2

In order to continue the installation process, please review the license
agreement.
Please, press ENTER to continue
```

Figure 7. Miniconda installation under root user

6) Let's install the dependencies now: run commands one by one - ***conda install numpy***, ***conda install scipy***, ***conda install gdal***.

These commands install necessary python packages in "base" conda environment. After successful installation of the packages you are ready to configure MAJA.

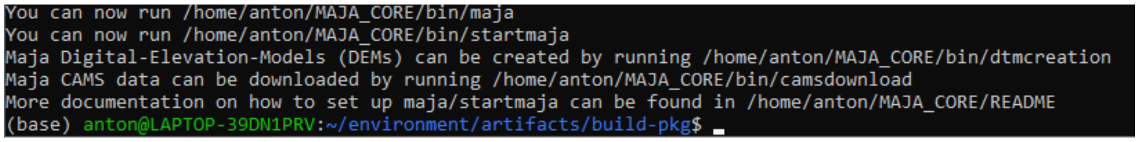
III. MAJA configuration.

1) Firstly, you need to obtain a copy of MAJA 4.2.1 which is available at CNES website from data section. ***Navigate to the website, click download tab, select version 4.2.1, fill general info, and download the archive.***

2) artifacts.zip is what you need, ***unzip it and take MAJA-4.2.1.run to the same place where Miniconda is.***

3) Check availability by running ***ls*** and if the file is there, run ***./MAJA-4.2.1.run --target /MAJA_CORE***

Step three performs installing MAJA into /MAJA_CORE directory and after successful installation the following text will appear (Figure 8):



```
You can now run /home/anton/MAJA_CORE/bin/maja
You can now run /home/anton/MAJA_CORE/bin/startmaja
Maja Digital-Elevation-Models (DEMs) can be created by running /home/anton/MAJA_CORE/bin/dtmcreation
Maja CAMS data can be downloaded by running /home/anton/MAJA_CORE/bin/camsdownload
More documentation on how to set up maja/startmaja can be found in /home/anton/MAJA_CORE/README
(base) anton@LAPTOP-39DN1PRV:~/environment/artifacts/build-pkg$
```

Figure 8. Successful MAJA installation

4) After all those steps MAJA in not ready yet, to make it finally able to run you need to create a folder.txt file in /MAJA_CORE/bin with the following content:

```
[Maja_Inputs]
repWork=/MAJA_JOBS/work
repGipp=/MAJA_JOBS/gipp
repMNT=/MAJA_JOBS/dtm
repL1  =/Input
repL2  =/Output
exeMaja=/MAJA_CORE/bin/maja
repCAMS=/path/to/CAMS

[DTM_Creation]
repRAW=/MAJA_JOBS/dtm/raw
repGSW=/MAJA_JOBS/dtm/gsw
```

This file serves as configuration file for MAJA, so it knows where to take data from and where save the results to. repWork is a directory to store the temporary files, repGipp is the folder where Start_maja automatically downloads the GIPP-set for each plugin, repMNT stores the DTM (MNT in french) in Maja format, repL1 is where to find the L1C data, repL2 is for the L2A data (without the site name which is added aferwards), exeMaja is the path to maja's executable file, repCAMS is where CAMS data is stored (not in use currently). repRAW stores the raw DTM archives, repGSW stores the raw Water-Mask files.

MAJA is now ready, and you need to provide some data for it.

IV. Data download and running maja.

A directory with information about satellite image is called *product* and it contains not only images itself, but also additional metadata information.

1) Since you are supposed to provide time-series data, it is possible to use CREODIAS finder to obtain necessary products. For that, ***navigate to CREODIAS website listed in the data section, register, log in and download the following products:***

S2B_MSIL1C_20210517T103619_N0300_R008_T31TFJ_20210517T124830.SAFE
S2B_MSIL1C_20210514T102559_N0300_R108_T31TFJ_20210514T123904.SAFE
S2A_MSIL1C_20210512T104021_N0300_R008_T31TFJ_20210512T130043.SAFE

Let's take a look at the important parts of the names: S2B and S2A stands for the satellite, but since Sentinel-2A and Sentinel-2B are technically identical, images can be interchanged/combined together. L1C in the product name tells that it is "Top-Of-Atmosphere (TOA)" image which means that no atmospheric corrections (e.g. discoloration removal, smog removal) have been performed. In addition to cloud mask generation, MAJA also deals with atmospheric corrections, so the result you get will be L2A product, which means "Bottom-Of-Atmosphere (BOA)" with removed atmospheric effects. Next, these products have the same index – T31TFJ meaning that they are from the same area, while dates are different: 12, 15 and 17th of May (first date is the observation date, second is the publication date).

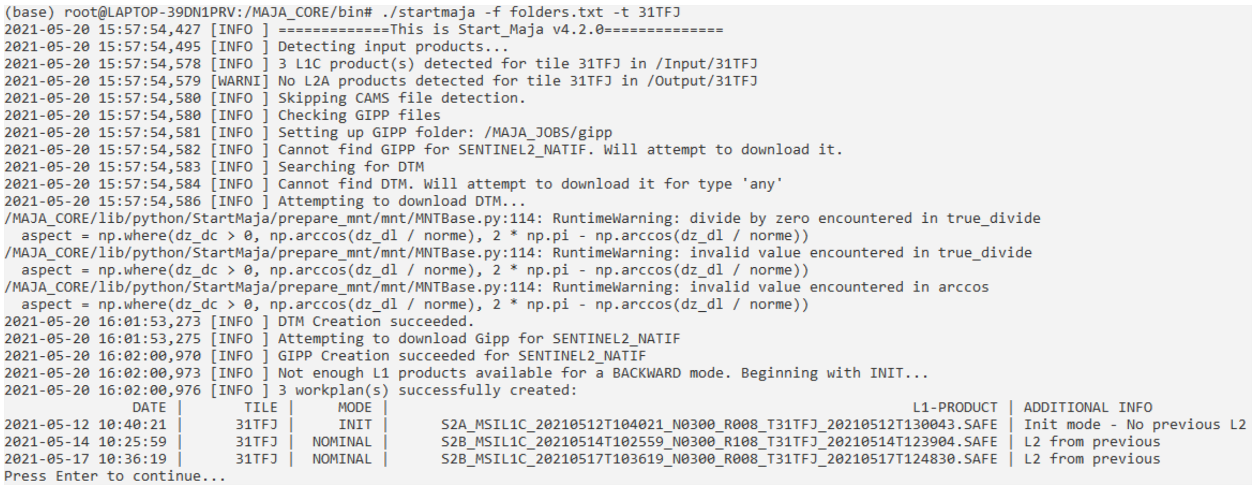
2) ***Downloaded products are supposed to be unzipped and .SAFE folders should be put in the repL1 directory (in this case /Input) under additional 31TFJ folder.*** The final structure will look like this:

/Input/31TFJ/S2B_MSIL1C_20210517T103619_N0300_R008_T31TFJ_20210517T124830.SAFE
/Input/31TFJ/S2B_MSIL1C_20210514T102559_N0300_R108_T31TFJ_20210514T123904.SAFE
/Input/31TFJ/S2A_MSIL1C_20210512T104021_N0300_R008_T31TFJ_20210512T130043.SAFE

3) In addition to time-series data, MAJA also requires digital elevation model. For that, you can ***navigate to Copernicus website, register, log in and download eu_dem_v11_E30N20.zip digital elevation model.*** eu_dem_v11_E30N20.zip should be located at repRAW which is /MAJA_JOBS/dtm/raw/eu_dem_v11_E30N20.zip

4) Time to run! Navigate to /MAJA_CORE/bin and execute ./startmaja -f folders.txt -t 31TFJ

After that MAJA starts to work and you should see how it performs:



```
(base) root@LAPTOP-39DN1PRV:/MAJA_CORE/bin# ./startmaja -f folders.txt -t 31TFJ
2021-05-20 15:57:54,427 [INFO ] =============This is Start_Maja v4.2.0==============
2021-05-20 15:57:54,495 [INFO ] Detecting input products...
2021-05-20 15:57:54,578 [INFO ] 3 L1C product(s) detected for tile 31TFJ in /Input/31TFJ
2021-05-20 15:57:54,579 [WARNI] No L2A products detected for tile 31TFJ in /Output/31TFJ
2021-05-20 15:57:54,580 [INFO ] Skipping CAMS file detection.
2021-05-20 15:57:54,580 [INFO ] Checking GIPP files
2021-05-20 15:57:54,581 [INFO ] Setting up GIPP folder: /MAJA_JOBS/gipp
2021-05-20 15:57:54,582 [INFO ] Cannot find GIPP for SENTINEL2_NATIF. Will attempt to download it.
2021-05-20 15:57:54,583 [INFO ] Searching for DTM
2021-05-20 15:57:54,584 [INFO ] Cannot find DTM. Will attempt to download it for type 'any'
2021-05-20 15:57:54,586 [INFO ] Attempting to download DTM...
/MAJA_CORE/lib/python/StartMaja/prepare_mnt/mnt/MNTBase.py:114: RuntimeWarning: divide by zero encountered in true_divide
  aspect = np.where(dz_dc > 0, np.arccos(dz_dl / norme), 2 * np.pi - np.arccos(dz_dl / norme))
/MAJA_CORE/lib/python/StartMaja/prepare_mnt/mnt/MNTBase.py:114: RuntimeWarning: invalid value encountered in true_divide
  aspect = np.where(dz_dc > 0, np.arccos(dz_dl / norme), 2 * np.pi - np.arccos(dz_dl / norme))
/MAJA_CORE/lib/python/StartMaja/prepare_mnt/mnt/MNTBase.py:114: RuntimeWarning: invalid value encountered in arccos
  aspect = np.where(dz_dc > 0, np.arccos(dz_dl / norme), 2 * np.pi - np.arccos(dz_dl / norme))
2021-05-20 16:01:53,273 [INFO ] DTM Creation succeeded.
2021-05-20 16:01:53,275 [INFO ] Attempting to download Gipp for SENTINEL2_NATIF
2021-05-20 16:02:00,970 [INFO ] GIPP Creation succeeded for SENTINEL2_NATIF
2021-05-20 16:02:00,973 [INFO ] Not enough L1 products available for a BACKWARD mode. Beginning with INIT...
2021-05-20 16:02:00,976 [INFO ] 3 workplan(s) successfully created:
               DATE |       TILE |     MODE |                                                        L1-PRODUCT | ADDITIONAL INFO
2021-05-12 10:40:21 |      31TFJ |     INIT |      S2A_MSIL1C_20210512T104021_N0300_R008_T31TFJ_20210512T130043.SAFE | Init mode - No previous L2
2021-05-14 10:25:59 |      31TFJ |  NOMINAL |      S2B_MSIL1C_20210514T102559_N0300_R108_T31TFJ_20210514T123904.SAFE | L2 from previous
2021-05-17 10:36:19 |      31TFJ |  NOMINAL |      S2B_MSIL1C_20210517T103619_N0300_R008_T31TFJ_20210517T124830.SAFE | L2 from previous
Press Enter to continue...
```

Figure 9. Successful MAJA initialization

On the figure 9 you can see that three workplans have been created. Note that the first product has INIT mode, while later ones are in NOMINAL mode. INIT means that the product is going to be processed not as a time series data, but rather as a separate one (see additional info column), while NOMINAL mode considers previous products taking the advantage of the time series data format. In this case, however, the quality of INIT product is very poor since it means no composite and only multispectral criteria is used. To avoid such a problem, you can use additional flag -- nbackward 2, to run the whole process in reverse time, so the first product will take advantage of later ones. The difference can be quite visible as it is shown on the picture below:

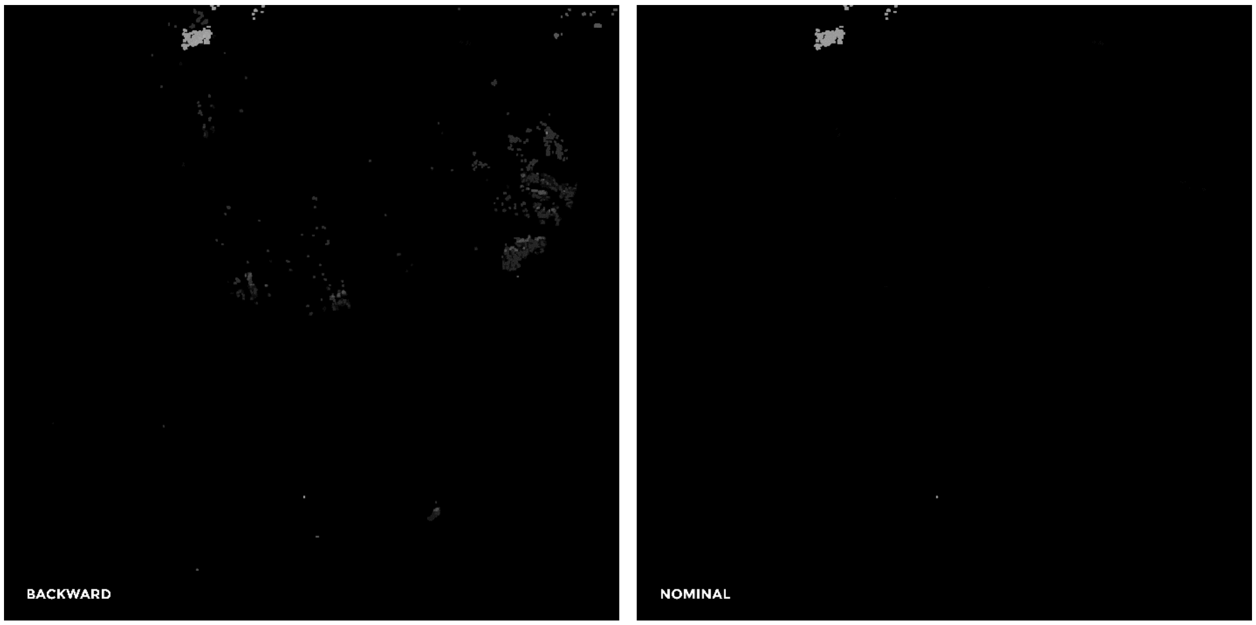


Figure 10. Difference between INIT and BACKWARD modes

5) Press enter and wait till the very end. Be patient, because it takes time for MAJA to process the data and the whole process might take around one hour, depending on your computer configuration.

It might be so that you encounter an issue related to libffi.so.6
Is happens because of the too new Ubuntu version and packages do not refer to a new file yet.
If it happened, run the following: ***find /usr/lib -name "libffi.so*"*** and then ***sudo ln -s /usr/lib/x86_64-linux-gnu/libffi.so.7 /usr/lib/x86_64-linux-gnu/libffi.so.6*** and retry ./startmaja

When MAJA is done, take a look at the results:

6) ***Navigate to /Output folder and find SENTINEL2B_20210514-103853-056_L2A_T31TFJ_C_V1-0 product***, which is the second in the time-series, ***open it -> MASKS -> open CLM_R1 mask in QGIS or ArcGIS software.***

After opening a mask, you will see different numerical classes (Figure 11):

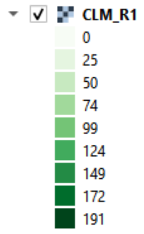

CLM_R1
0
25
50
74
99
124
149
172
191

Figure 11. Generated classes for cloudmask

Those values are a part of THEIA MUSCATE L2A format from processing center at CNES. R1 in the name stands for 10 meters resolution, R2 mask (that can be found in the same folder) – 20 meters resolution. It is different from data format used in ESA, but there is an extensive documentation that explains it. Since you are interested in a cloud mask, let's see how it works.

Table 1. Cloud mask bits

bit 0 (1)	all clouds except the thinnest and all shadows
bit 1 (2)	all clouds (except the thinnest)
bit 2 (4)	clouds detected via mono-temporal thresholds
bit 3 (8)	clouds detected via multi-temporal thresholds
bit 4 (16)	thinnest clouds
bit 5 (32)	cloud shadows cast by a detected cloud
bit 6 (64)	cloud shadows cast by a cloud outside image
bit 7 (128)	high clouds detected by 1.38 µm

In the table 1 you can see different bits from 0 to 7 that represent particular feature that is detected using particular method. It may noticed already that values in CLM_R1 mask are different, but the reason for that is because they incorporate different features, here is the example:

*Example 1: value 33 = 32+1 (00**1**0000**1**): bit 0 tells that a shadow or a cloud has been detected and bit 5 that it is a shadow.*

If to decipher your values, here is what we can get:

Example 2: value 25 = 16+8+1 (00011001): bit 0 tells that a shadow or a cloud has been detected, bit 3 that a cloud has been detected via multi-temporal thresholds, bit 4 tells that it is a thinnest cloud.

Example 3: value 74 = 64+8+2 (01001010): bit 1 tells that a cloud has been detected, bit 3 that a cloud has been detected via multi-temporal thresholds, bit 6 tells that cloud is outside the image.

Classifying values this way provides a detailed segmentation of the cloud mask making it possible to use for specific tasks, but here you are going to use a “Strict Cloud Mask”. For that, you do not need to classify the values and everything except 0 values can be considered as a cloud or cloud shadow that represent areas which potentially can be misclassified.

7) Underneath cloud mask ***put L1C image of the same date*** from .SAFE directory under GRANULE/<PRODUCT_INDEX>/IMG_DATA/ TCI band image.

8) Set up an appropriate symbology for the layers and obtain the final result:

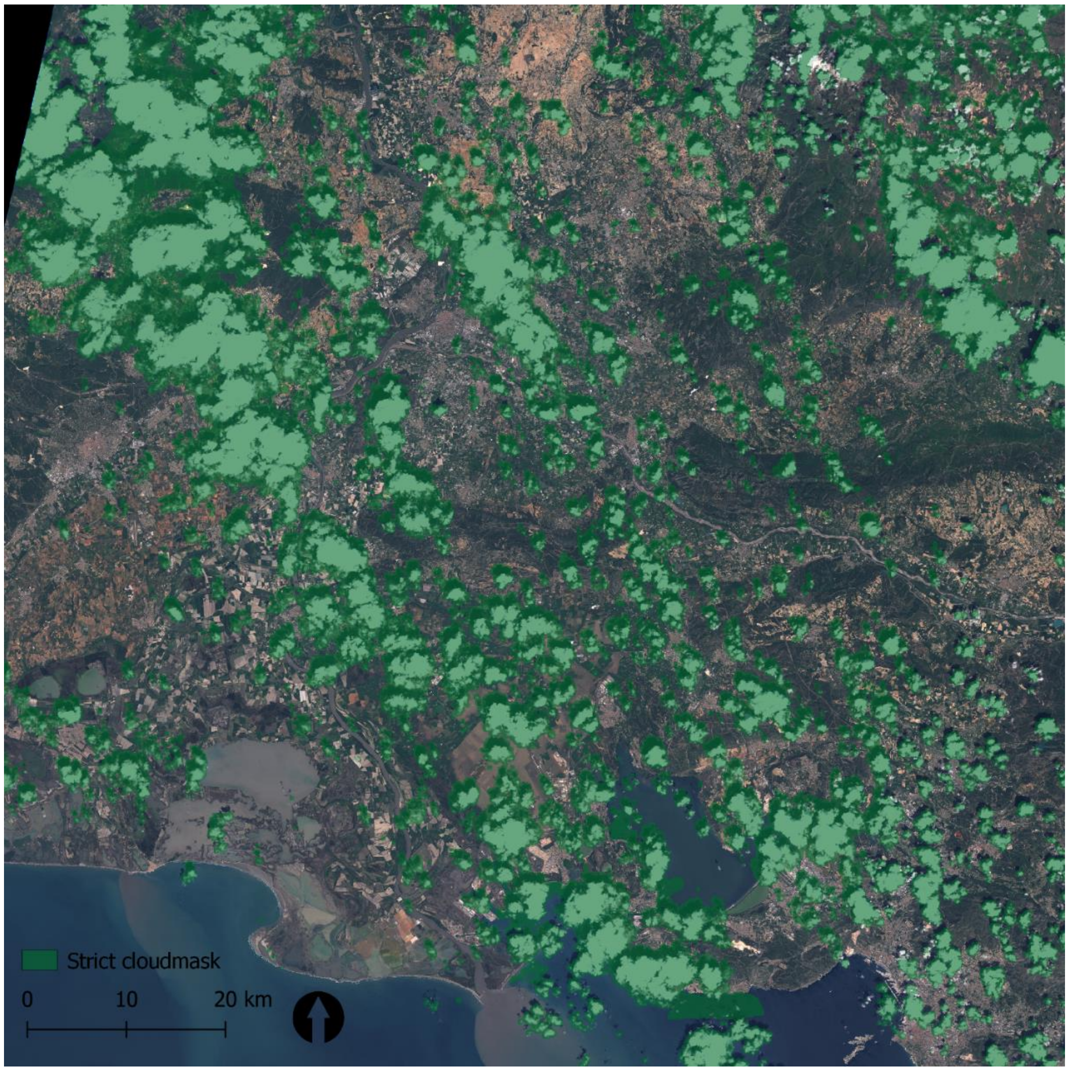


Figure 12. Final result map

References.

Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y., 2015. Time-series clustering – A decade review. *Information Systems*, Volume 53, 16-38. https://doi.org/10.1016/j.is.2015.04.007

Ansari, M.Y., Ahmad, A., Khan, S.S. et al., 2020. Spatiotemporal clustering: a review. *Artif Intell* Rev 53, 2381–2423. https://doi.org/10.1007/s10462-019-09736-1