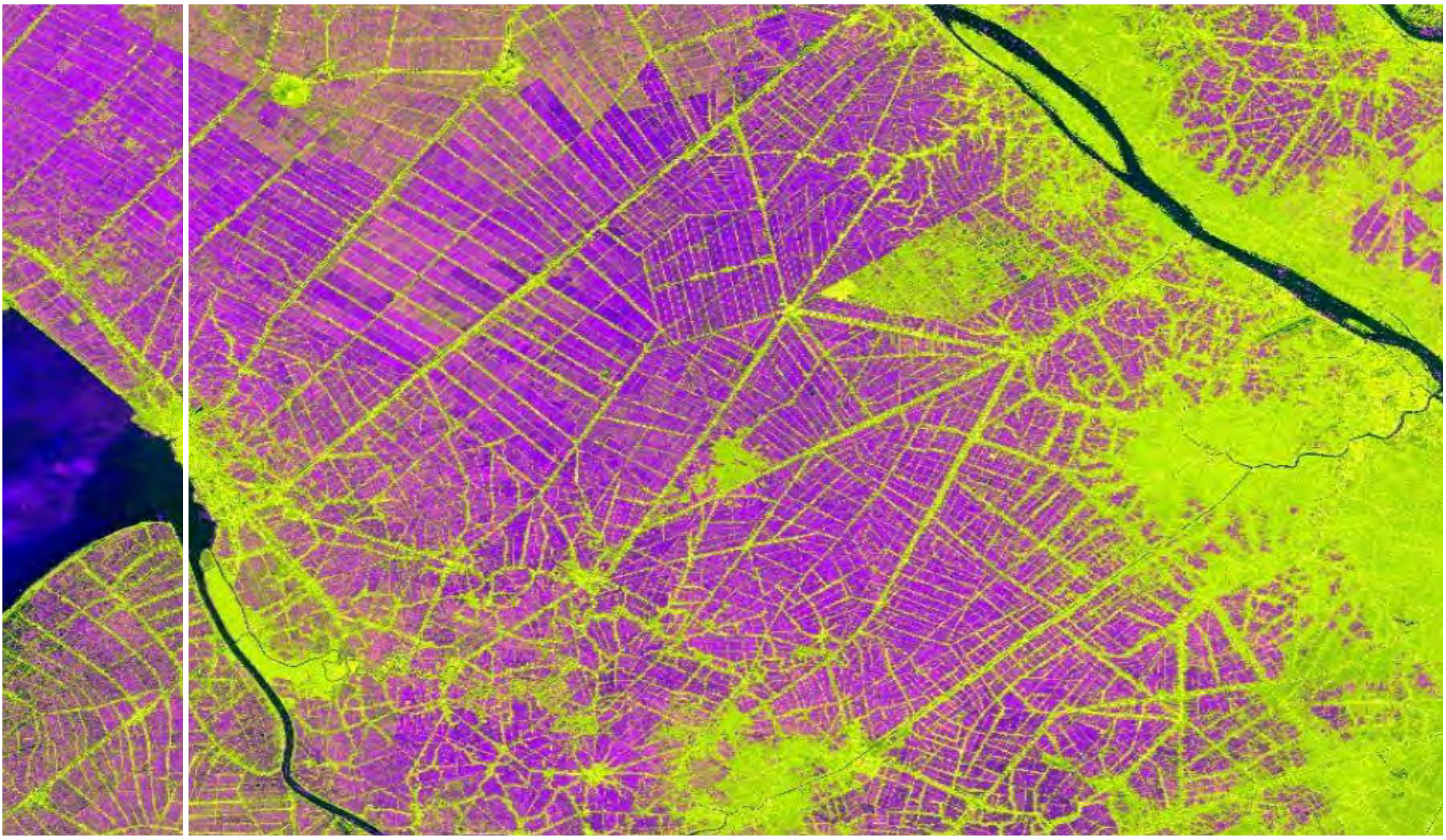


TRAINING KIT – LAND10

RICE MAPPING WITH SENTINEL-1
Case Study: Vietnam, 2018

Research and User Support for Sentinel Core Products

The RUS Service is funded by the European Commission, managed by the European Space Agency and operated by CSSI and its partners.

Authors would be glad to receive your feedback or suggestions and to know how this material was used. Please, contact us on training@rus-copernicus.eu

Cover images produced by RUS Copernicus

The following training material has been prepared by Serco Italia S.p.A. within the RUS Copernicus project.

Date of publication: April 2019

Version: 1.1

Suggested citation:

Serco Italia SPA (2019). *Rice Mapping with Sentinel-1 – Case Study: Vietnam 2018 (version 1.2).* Retrieved from RUS Lectures at https://rus-copernicus.eu/portal/the-rus-library/learn-by-yourself/

Table of Contents

1 Introduction to RUS 4
2 Rice mapping – background 4
3 Training 4
3.1 Data used 4
3.2 Software in RUS environment 4
4 Register to RUS Copernicus 5
5 Request a RUS Copernicus Virtual Machine 6
6 Step by step 10
6.1 Data download – ESA SciHUB 10
6.2 Download data 11
6.3 Sentinel-1 SNAP GPT Preprocessing 13
6.3.1 Read 14
6.3.2 Apply orbit file 14
6.3.3 Thermal Noise Removal 15
6.3.4 Calibration 15
6.3.5 Terrain correction 16
6.3.6 Subset 17
6.3.7 Write 18
6.3.8 Graph Processing Tool | GPT 18
6.3.9 Batch processing 21
6.3.10 Stack 25
6.3.11 Multi-temporal speckle filtering 26
6.3.12 Decibel transformation 27
6.4 QGIS 28
6.4.1 Sigma Naught evolution 28
6.4.2 QGIS Processing 30
7 Further reading and resources 34

1 Introduction to RUS

The Research and User Support for Sentinel core products (RUS) service provides a free and open scalable platform in a powerful computing environment, hosting a suite of open source toolboxes pre-installed on virtual machines, to handle and process data derived from the Copernicus Sentinel satellites constellation.

In this tutorial, we will employ RUS to detect rice using a dense time-series of Sentinel-1 GRD products as input data over an area in Vietnam.

2 Rice mapping – background

Rice fields in the south of Spain

According to the International Rice Research Institute (IRRI), rice, wheat, and maize are the three leading food crops in the world; together they directly supply more than 50% of all calories consumed by the entire human population. Wheat is the leader in area harvested each year with 214 million ha, followed by rice with 154 million ha and maize with 140 million ha. Human consumption accounts for 85% of total production for rice, compared with 72% for wheat and 19% for maize.

Rice is also the most important crop to millions of small farmers who grow it on millions of hectares throughout the region, and to the many landless workers who derive income from working on these farms.

A few years ago, the European Union (EU) started an ambitious program, Copernicus, which includes the launch of a new family of earth observation satellites known as Sentinels. Amongst other applications, this new generation of satellites will improve the identification, mapping, assessment, and monitoring of crops and their dynamics at a range of spatial and temporal resolutions.

3 Training

Approximate duration of this training session is **two** hour.

The Training Code for this tutorial is LAND10. If you wish to practice the exercise described below within the RUS Virtual Environment, register on the RUS portal and open a User Service request from Your RUS service > Your dashboard.

3.1 Data used

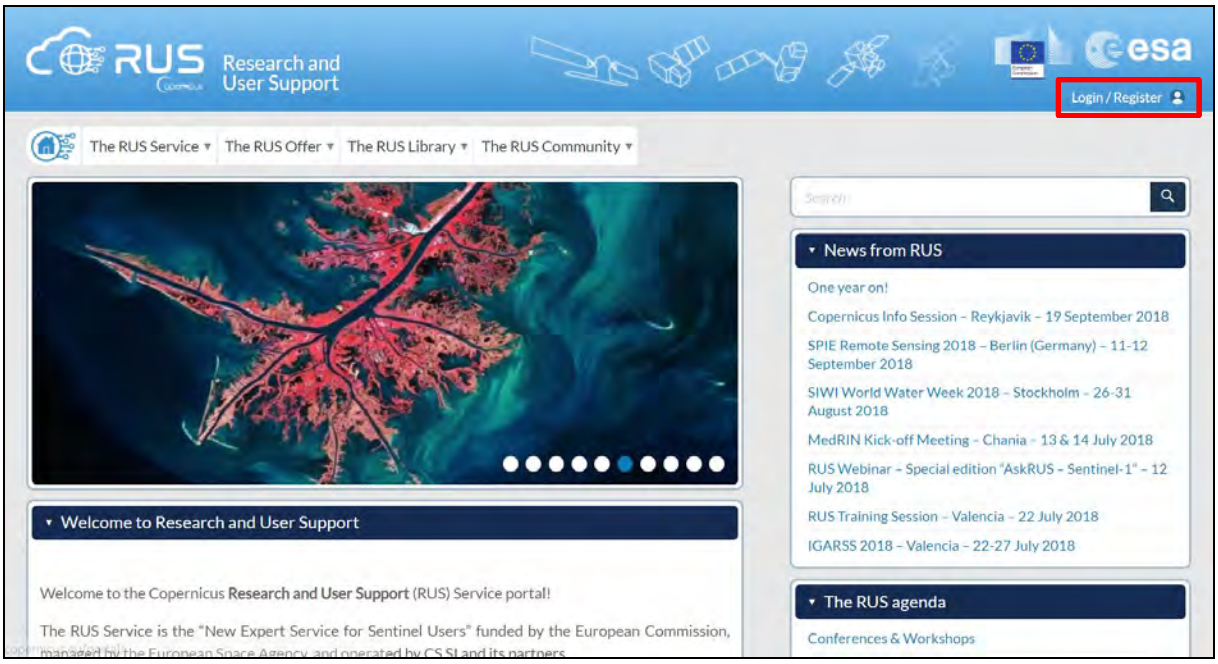
- 24 Sentinel-1A images acquired from March until December 2018 [downloadable at https://scihub.copernicus.eu/ using the .meta4 file provided in the *Original* folder of this exercise]
- Pre-processed data stored locally *@/shared/Training/LAND10_RiceMapping_Vietnam/AuxData/*

3.2 Software in RUS environment

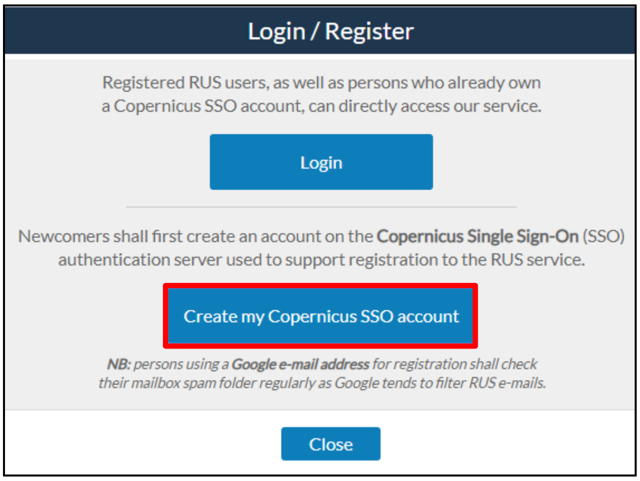
Internet browser, SNAP + GPT + S1 Toolbox + QGIS

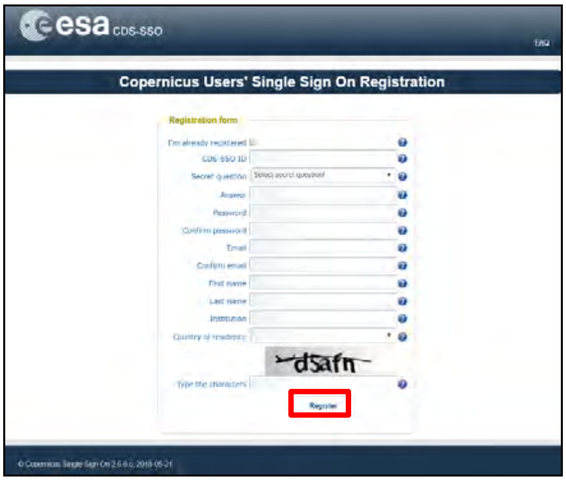
4 Register to RUS Copernicus

To repeat the exercise using a RUS Copernicus Virtual Machine (VM), you will first have to register as a RUS user. For that, go to the RUS Copernicus website (www.rus-copernicus.eu) and click on ***Login/Register*** in the upper right corner.



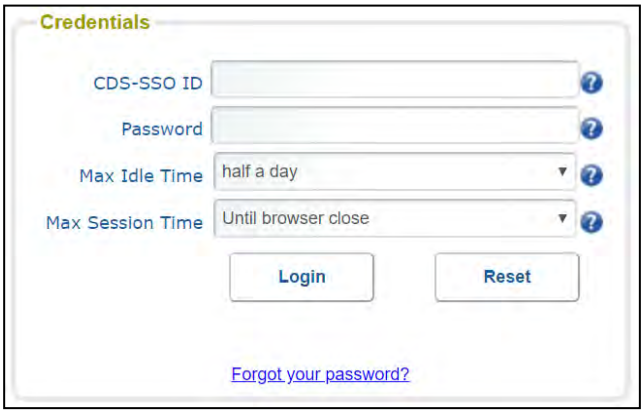
Select the option ***Create my Copernicus SSO account*** and then fill in ALL the fields on the **Copernicus Users' Single Sign On Registration**. Click ***Register***.



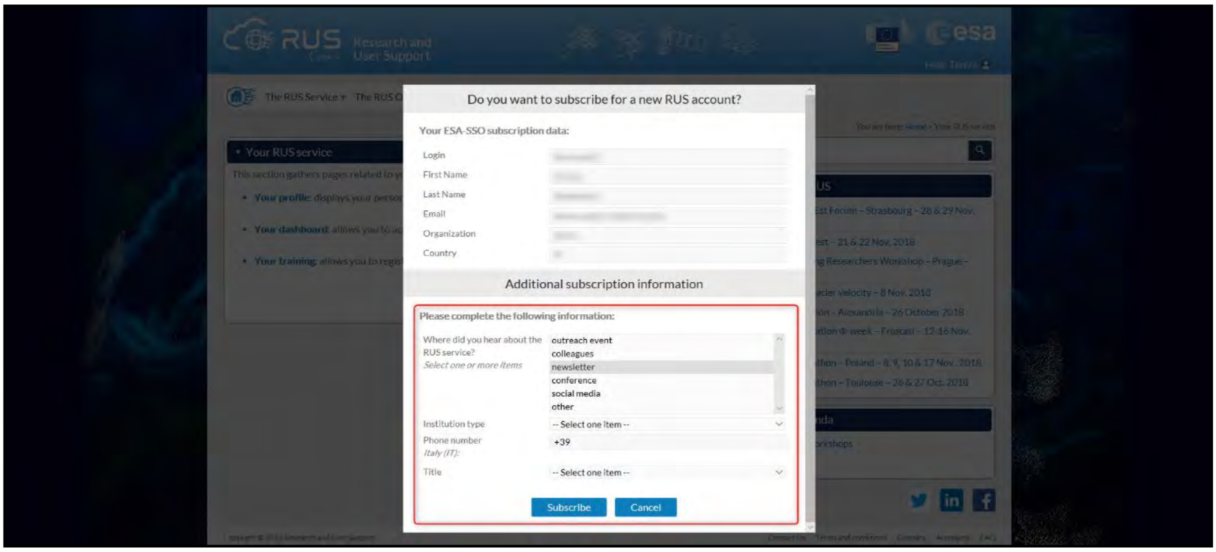


Within a few minutes you will receive an e-mail with activation link. Follow the instructions in the e-mail to activate your account.

You can now return to https://rus-copernicus.eu/, click on ***Login/Register***, choose ***Login*** and enter your chosen credentials.

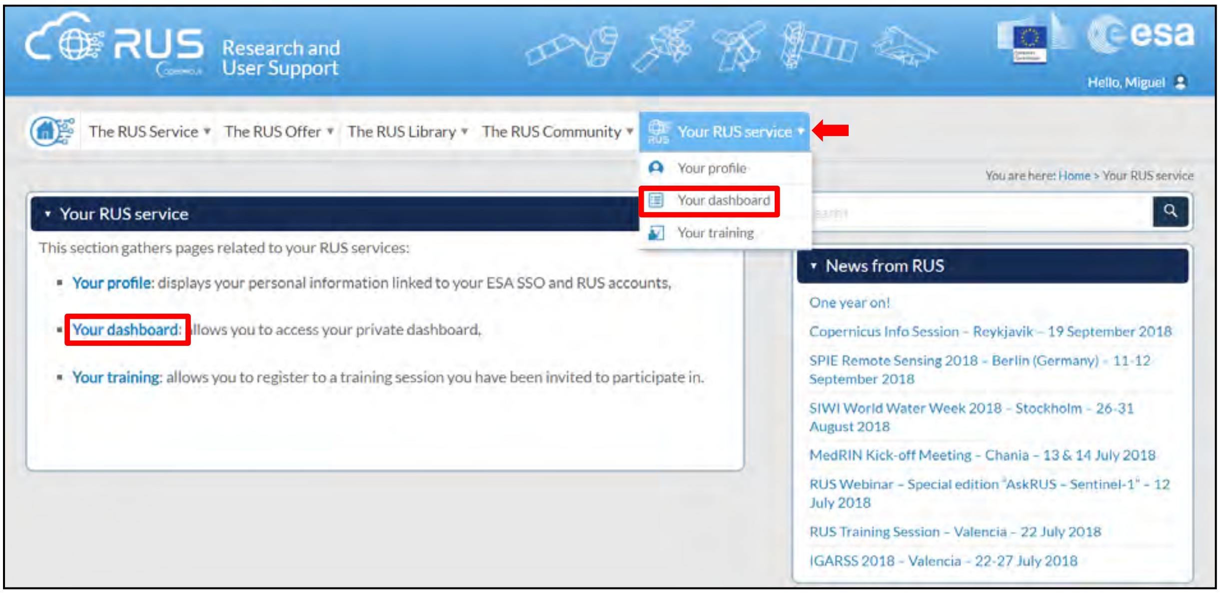


Upon your first login you will need to enter some details. You must fill all the fields.

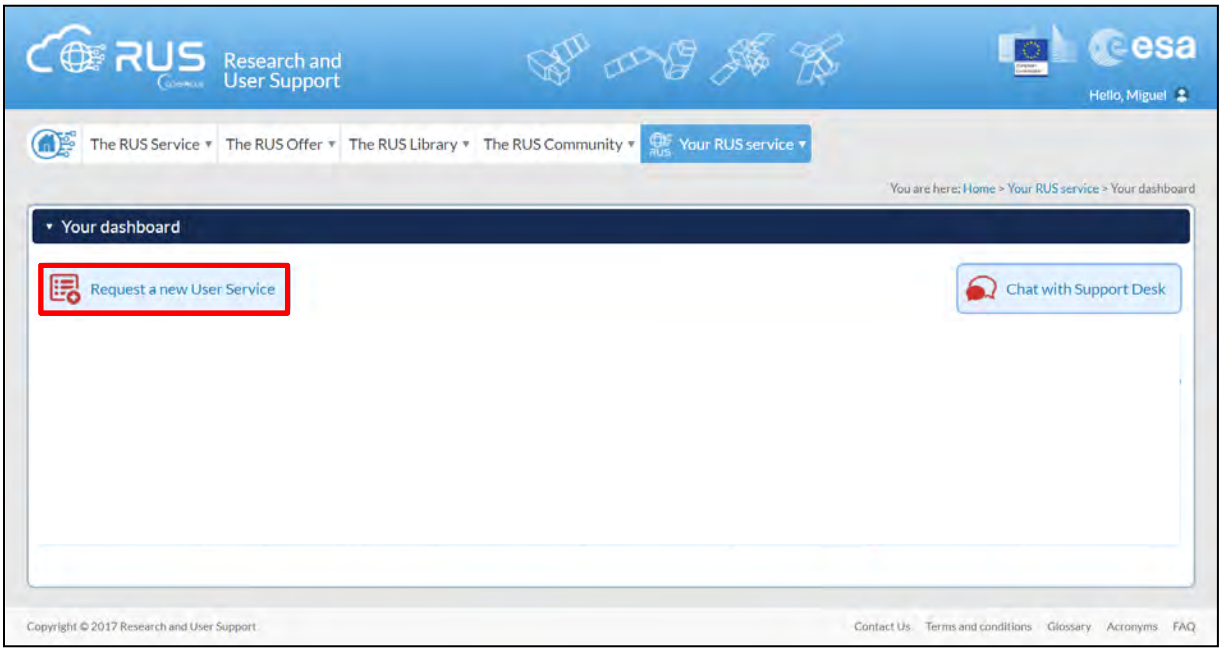


5 Request a RUS Copernicus Virtual Machine

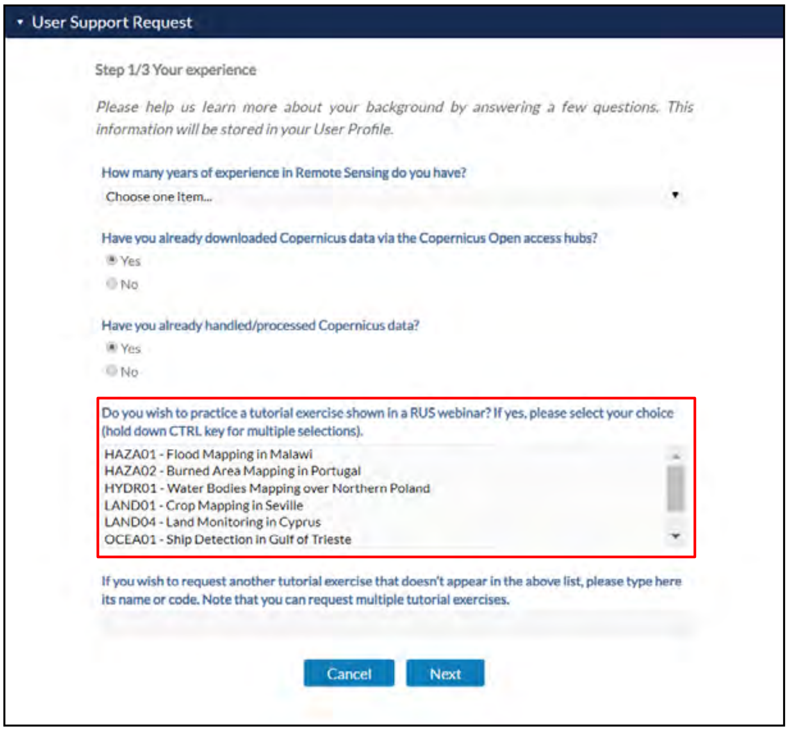
Once you are registered as a RUS user, you can request a RUS Virtual Machine to repeat this exercise or work on your own projects using Copernicus data. For that, log in and click on ***Your RUS Service → Your Dashboard***.



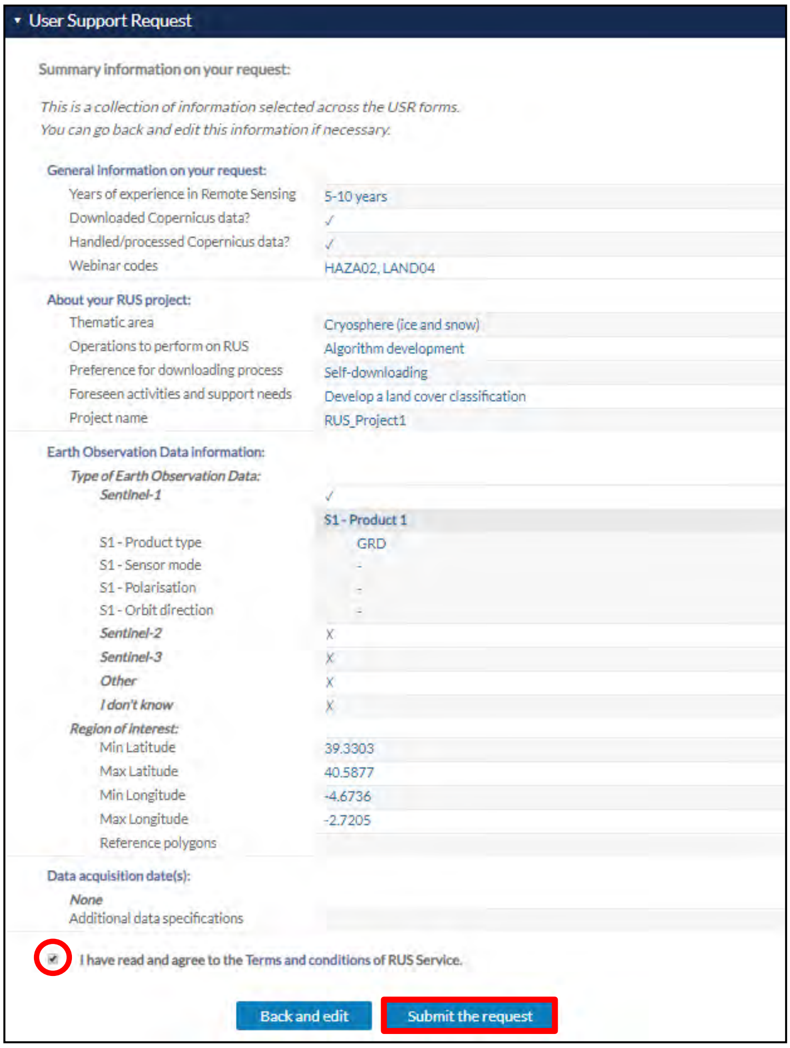
Click on ***Request a new User Service*** to request your RUS Virtual Machine. Complete the form so that the appropriate cloud environment can be assigned according to your needs.



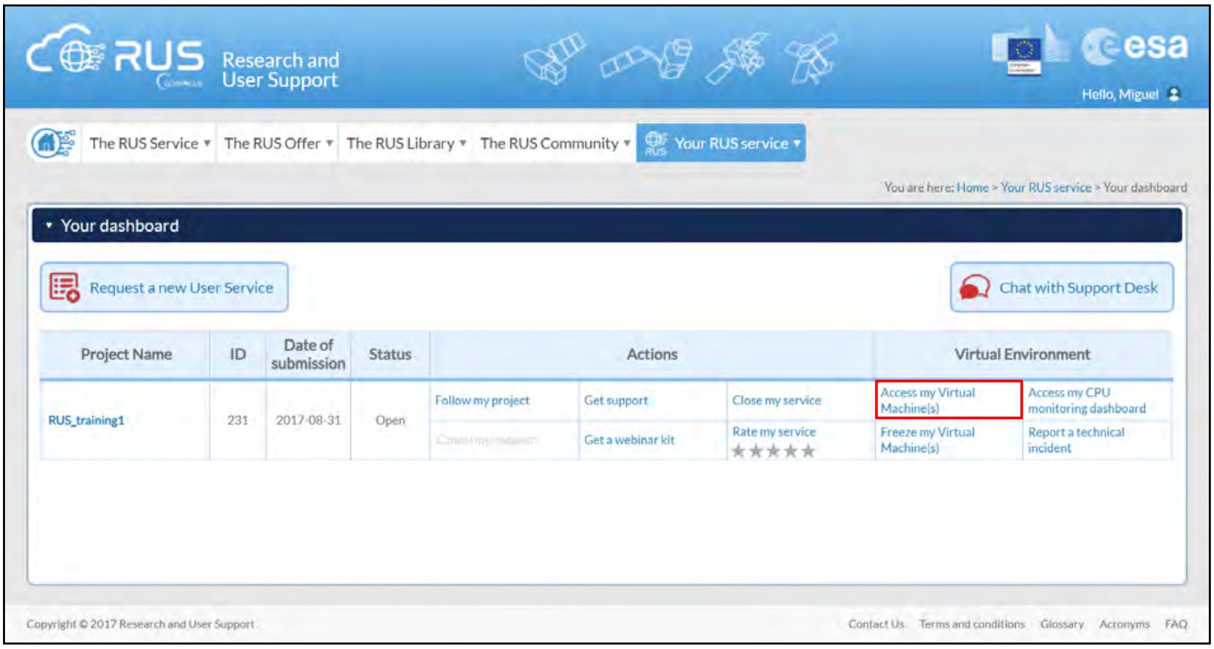
If you want to repeat this tutorial (or any previous one) select the one(s) of your interest in the appropriate field.



Complete the remaining steps, check the terms and conditions of the RUS Service and submit your request once you are finished.



Further to the acceptance of your request by the RUS Helpdesk, you will receive a notification email with all the details about your Virtual Machine. To access it, go to ***Your RUS Service → Your Dashboard*** and click on ***Access my Virtual Machine***.



Fill in the login credentials that have been provided to you by the RUS Helpdesk via email to access your RUS Copernicus Virtual Machine.

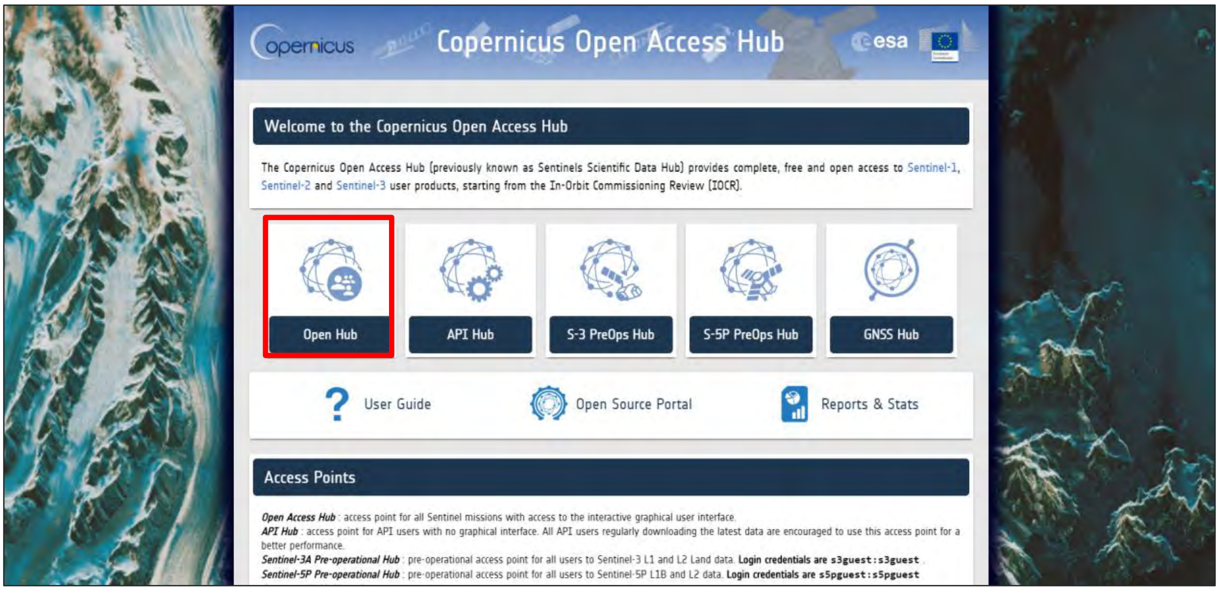
This is the remote desktop of your Virtual Machine.

6 Step by step

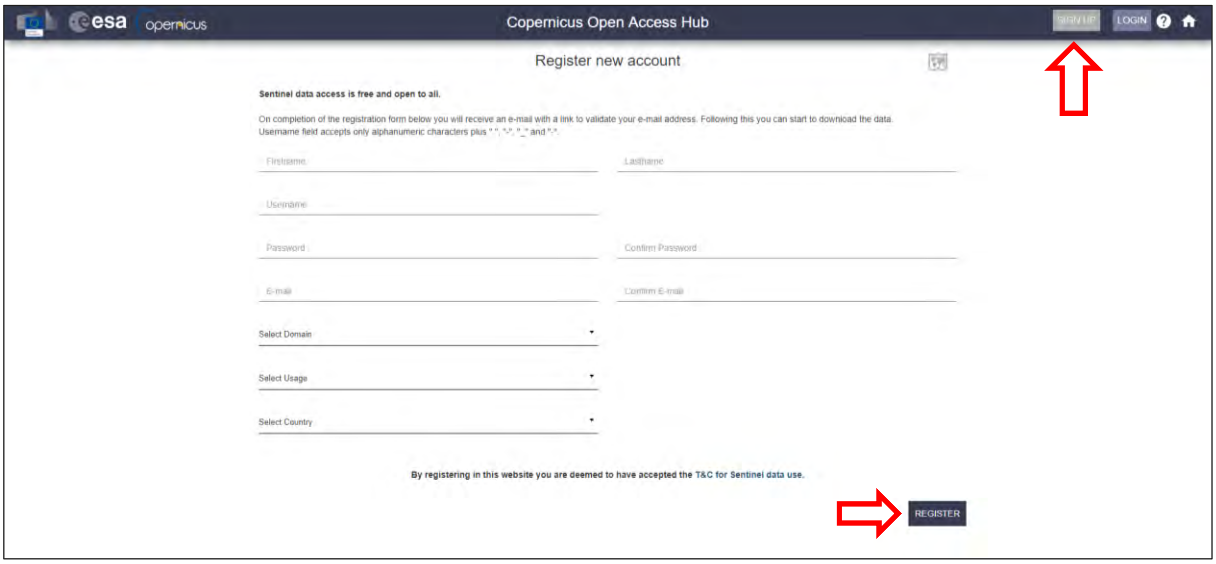
6.1 *Data download – ESA SciHUB*

Before starting the exercise, make sure you are registered in the Copernicus Open Access Hub so that you can access the free data provided by the Sentinel satellites.

Go to https://scihub.copernicus.eu/



Go to *Open Hub*. If you do not have an account, sign up in the upper right corner, fill in the details and click register.



You will receive a confirmation email on the e-mail address you have specified: open the email and click on the link to finalize the registration.

Once your account is activated – or if you already have an account – log in.

6.2 Download data

In this exercise, we will analyze 24 Sentinel-1A images during 2018. The following table shows the date and reference of the images that will be used:

SATELLITE	DATE	IMAGE ID
Sentinel-1A	2018-03-07	S1A_IW_GRDH_1SDV_20180307T224528_20180307T224553_020915_023E2F_F92F
	2018-03-19	S1A_IW_GRDH_1SDV_20180319T224528_20180319T224553_021090_0243B4_2FEF
	2019-03-31	S1A_IW_GRDH_1SDV_20180331T224528_20180331T224553_021265_024942_546B
	2018-04-12	S1A_IW_GRDH_1SDV_20180412T224529_20180412T224554_021440_024EB3_0516
	2018-04-24	S1A_IW_GRDH_1SDV_20180424T224529_20180424T224554_021615_025428_1B32
	2018-05-06	S1A_IW_GRDH_1SDV_20180506T224530_20180506T224555_021790_0259B4_0789
	2018-05-18	S1A_IW_GRDH_1SDV_20180518T224530_20180518T224555_021965_025F45_05AD
	2018-05-30	S1A_IW_GRDH_1SDV_20180530T224531_20180530T224556_022140_0264E6_8978
	2018-06-11	S1A_IW_GRDH_1SDV_20180611T224532_20180611T224557_022315_026A59_BBF6
	2018-06-23	S1A_IW_GRDH_1SDV_20180623T224532_20180623T224557_022490_026F94_1CA0
	2018-07-05	S1A_IW_GRDH_1SDV_20180705T224533_20180705T224558_022665_0274AF_03A1
	2018-07-17	S1A_IW_GRDH_1SDV_20180717T224534_20180717T224559_022840_027A09_D025
	2018-08-10	S1A_IW_GRDH_1SDV_20180810T224535_20180810T224600_023190_02850C_4510
	2018-08-22	S1A_IW_GRDH_1SDV_20180822T224536_20180822T224601_023365_028AB2_F68E
	2018-09-03	S1A_IW_GRDH_1SDV_20180903T224537_20180903T224602_023540_029043_3D7A
	2018-09-15	S1A_IW_GRDH_1SDV_20180915T224537_20180915T224602_023715_0295DF_3261
	2018-09-27	S1A_IW_GRDH_1SDV_20180927T224537_20180927T224602_023890_029B8C_2691
	2018-10-09	S1A_IW_GRDH_1SDV_20181009T224538_20181009T224603_024065_02A150_C5FB
	2018-10-21	S1A_IW_GRDH_1SDV_20181021T224538_20181021T224603_024240_02A700_E03C
	2018-11-02	S1A_IW_GRDH_1SDV_20181102T224538_20181102T224603_024415_02ACCD_DD65
	2018-11-14	S1A_IW_GRDH_1SDV_20181114T224537_20181114T224602_024590_02B33B_A620
	2018-11-26	S1A_IW_GRDH_1SDV_20181126T224537_20181126T224602_024765_02B9AA_C6AC
	2018-12-08	S1A_IW_GRDH_1SDV_20181208T224536_20181208T224601_024940_02BF81_757D
	2018-12-20	S1A_IW_GRDH_1SDV_20181220T224536_20181220T224601_025115_02C5D5_0E2A

To improve the data acquisition process, we will use a download manager (See NOTE 1) that will take care of downloading all products that will be used. The metadata of the Sentinel products are saved in a *products.meta4* file created using the 'Cart' option of the Copernicus Open Access Hub.

> NOTE 1: A download manager is a computer program dedicated to the task of downloading possibly unrelated stand-alone files from (and sometimes to) the Internet for storage. For this exercise, we will use aria2. Aria2 is a lightweight multi-protocol & multi-source command-line download utility. More info at: https://aria2.github.io/

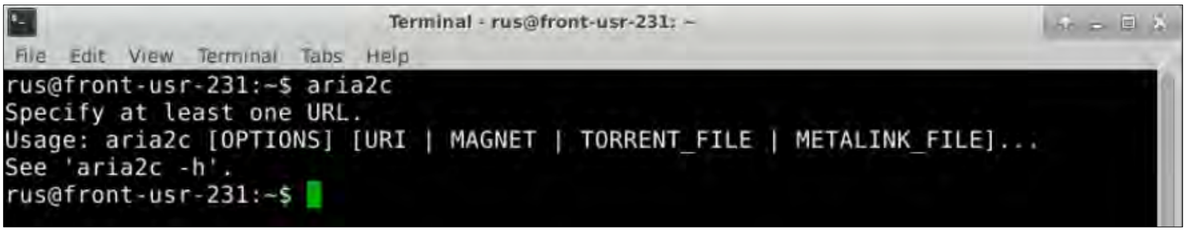
The *products.meta4* file containing the links to the Sentinel-1 products to be downloaded can be created following the methodology explained in NOTE 2. Follow the instructions and create your cart file, download it and save it in the following path:

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Original/*

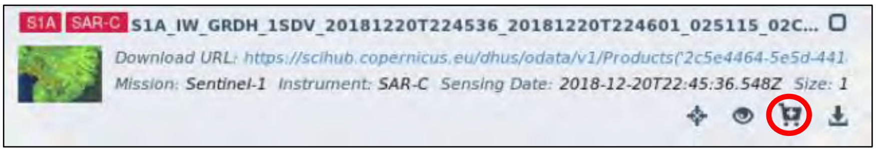
Before using the downloading manager and the .meta4 file, let's test if *aria2* is properly installed in the Virtual Machine. To do this, open the Command Line (in the bottom of your desktop window) and type the following and press *Enter:*

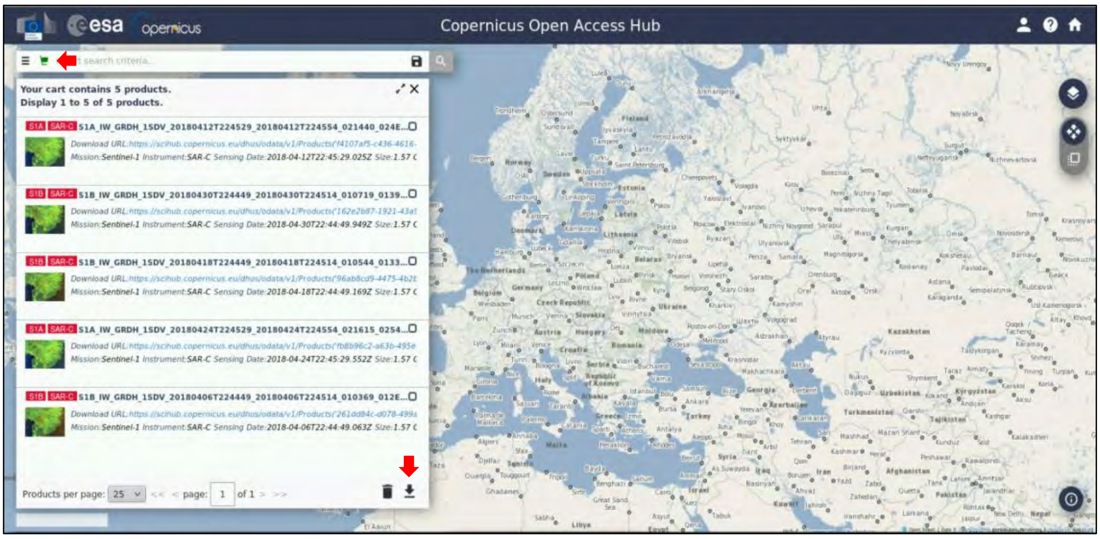
```
aria2c
```

If *aria2* is properly installed, the response should be as follows. If the response is '*-bash aria2c: command not found'* it means *aria2* is not installed (See NOTE 3).



NOTE 2: The Copernicus Open Access Hub allows you to add products to a 'Cart'. For that, perform a query; select the desired products from the result list and click on the 'Add Product to Cart' icon - . To find the appropriate images, copy-paste the image ID specified in the table (pg. 11) in the search box of the Copernicus Open Access Hub.



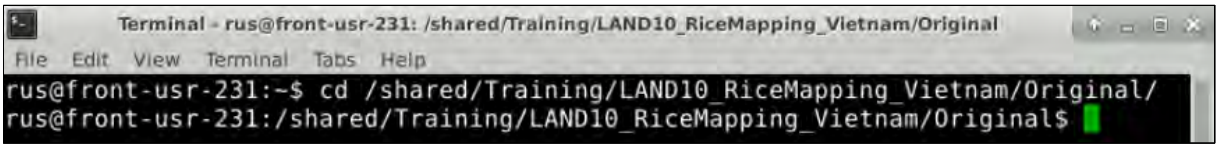


To view the products present in the cart just click anytime car icon - - on the top left corner of the screen. To download the cart click on "Download Cart" on the bottom right of the page. A download window will pop up, asking the user confirmation to save a .meta4 file named *'products.meta4'*. This file contains all the metalinks of the products.

NOTE 3: If **(and only if)** the response is '*-bash aria2c: command not found',* you need to install aria2. In the command line, type: *sudo apt-get install aria2*
When requested, type: Y
Once finished, test the installation as explained before.

Once *aria2* is ready to use, we can start the download process. For that, we need to navigate to the folder where the *products.meta4* file is stored. Type the following command in the terminal and run it (press *Enter*).

```
cd /shared/Training/LAND10_RiceMapping_Vietnam/Original/
```



Next, type the following command (in a single line) to run the download tool. Replace *username* and *password* (keep the quotation marks) with your login credentials for Copernicus Open Access Hub (COAH). Do not clear your cart in the COAH until the download process is finished.

```
aria2c --http-user='username' --http-passwd='password' --check-certificate=
false --max-concurrent-downloads=2 -M products.meta4
```

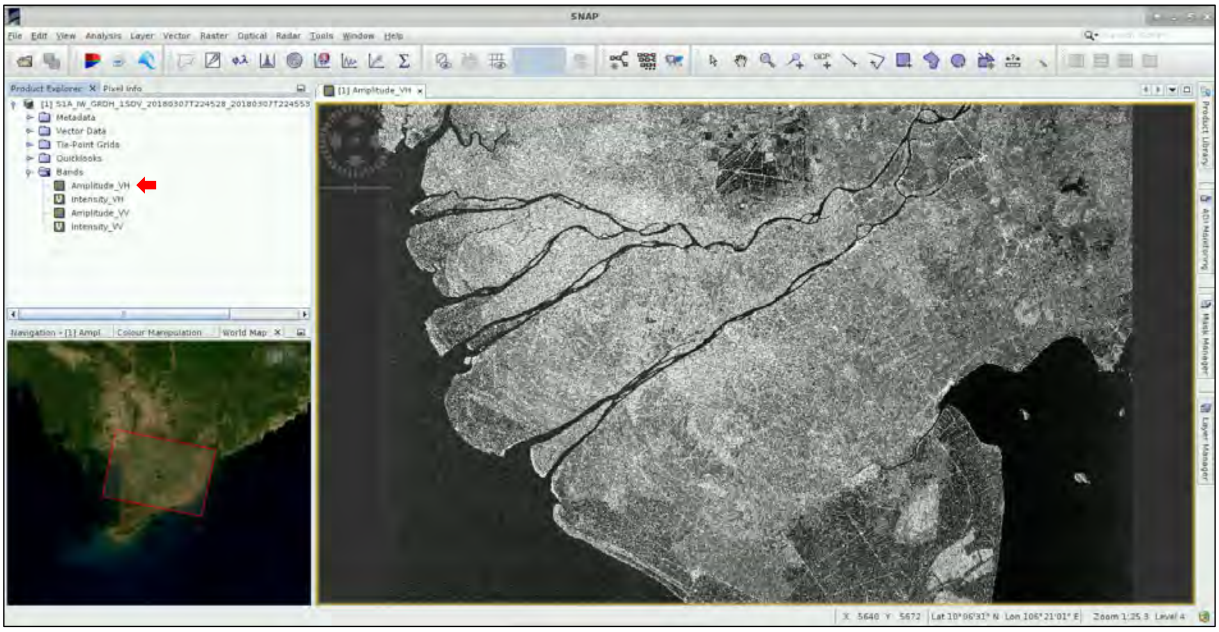
The Sentinel products will be saved in the same path where the *products.meta4* file is stored.

6.3 Sentinel-1 SNAP GPT Preprocessing

Once the Sentinel-1 images are downloaded, we need to run some pre-processing steps before they can be used for our final purpose. For this, we will use the SNAP software. In *Applications -> Processing* open **SNAP Desktop**; click **Open product**, navigate to the following path and open the first S1 image (2018-03-07).

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Original/*

The opened product will appear in Product Explorer. Click + to expand the contents of the first image, then expand the *Bands* folder and double click on *Amplitude_VH* to visualize it. (See NOTE 4).

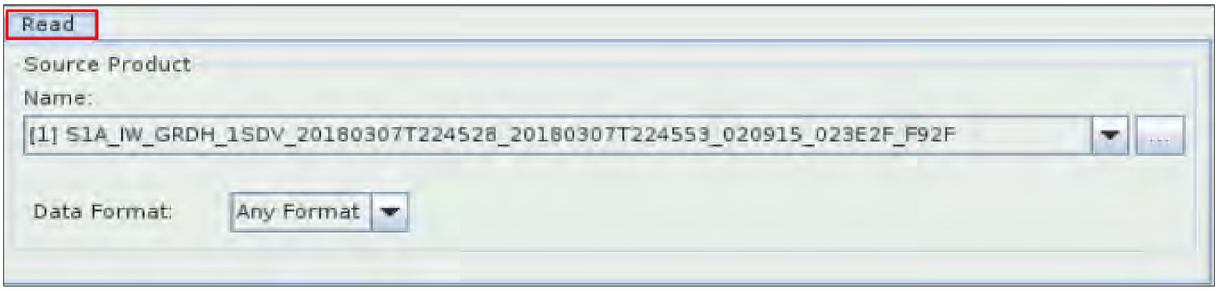


To process this and the other Sentinel-1 images, we will take advantage of the batch processing option available in SNAP GPT. In this way, we can define a specific processing chain and apply it to several images in an automatic way. This allows reducing processing time and storage requirement since no intermediate steps are created. Only the final product is physically saved.

Before running batch processing, it is necessary to create a graph containing all the processing steps. Go to *Tools -> Graph Builder.* So far, the graph only has two operators: *Read* (to read the input) and *Write* (to write the output). By right-clicking on the white space at the top panel, you can add an operator while a corresponding tab is created and added at the bottom panel. To avoid confusion, right click on the *Write* operator and delete it.

6.3.1 Read

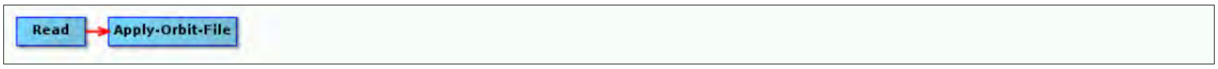
By default, Graph Builder will use the image we have previously opened in SNAP as the input image for the processing chain. Since we will be changing this parameter later in GPT, there is no need to change anything.

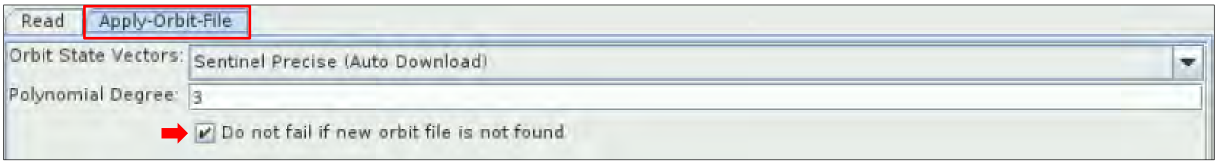


6.3.2 Apply orbit file

The first step of our Sentinel-1 pre-processing chain will update the orbit metadata (See NOTE 4) of the product to provide accurate satellite position and velocity information. To add the operator to our graph, right click and navigate to *Add -> RADAR -> Apply-Orbit-File*. Connect the new *Apply-Orbit-File* operator with the *Read* operator by clicking to the right side of the *Read* operator and dragging the red arrow towards the *Apply-Orbit-File* operator. In the corresponding tab, check the option *Do not fail if new orbit file is not found.*

> NOTE 4: The orbit state vectors provided in the metadata of a SAR product are generally not accurate and can be refined with the precise orbit files, which are available days-to-weeks after the generation of the product. The orbit file provides accurate satellite position and velocity information. Based on this information, the orbit state vectors in the abstract metadata of the product are updated. (*SNAP Help*)

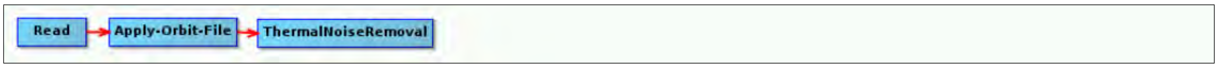




6.3.3 Thermal Noise Removal

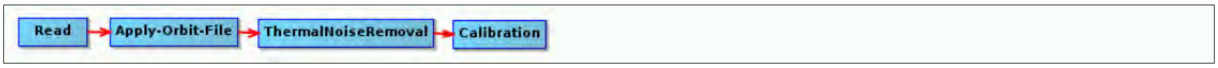
Next, we will remove the thermal noise (See NOTE 5). To add the operator to our graph, right click and navigate to *Add -> RADAR -> Radiometric -> ThermalNoiseRemoval*. In the corresponding tab, leave all the parameters for this operator as default. Connect the *ThermalNoiseRemoval* operator with the *Apply-Orbit-File* operator by clicking to the right side of the *Apply-Orbit-File* operator and dragging the red arrow towards the *ThermalNoiseRemoval* operator.

> NOTE 5: Thermal noise in SAR imagery is the background energy that is generated by the receiver itself. (*SNAP Help*) It skews the radar reflectivity to towards higher values and hampers the precision of radar reflectivity estimates. Level-1 products provide a noise LUT for each measurement dataset, provided in linear power, which can be used to remove the noise from the product.

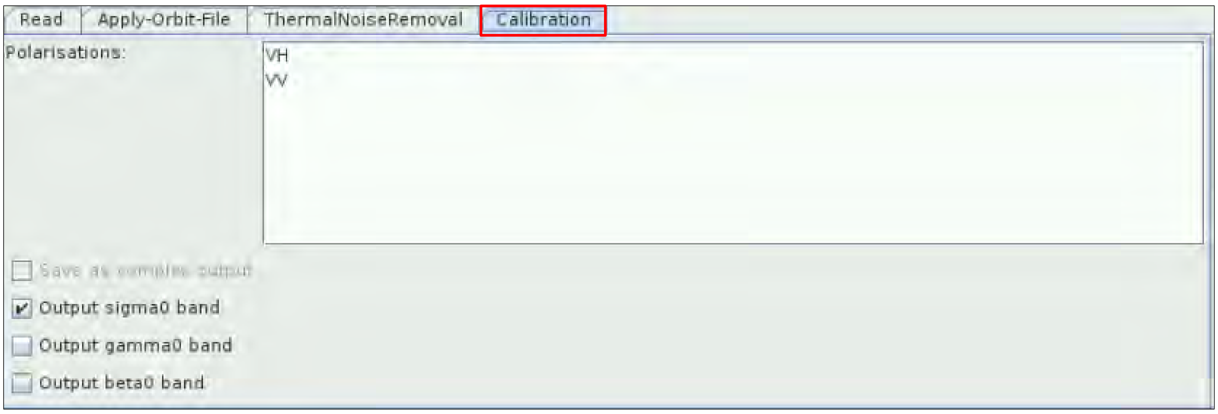


6.3.4 Calibration

Now, we can perform the Radiometric calibration. The objective of SAR calibration is to provide imagery in which the pixel values can be directly related to the radar backscatter. Though uncalibrated SAR imagery is sufficient for qualitative use, calibrated SAR images are essential to quantitative use of SAR data (See NOTE 6). To add the operator to our graph, right click and navigate to *Add -> RADAR -> Radiometric -> Calibration*. In the corresponding tab, leave all the parameters for this operator as default. Connect the *Calibration* operator with the *ThermalNoiseRemoval* operator by clicking to the right side of the *ThermalNoiseRemoval* operator and dragging the red arrow towards the *Calibration* operator.



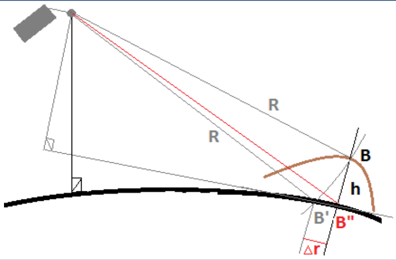
> NOTE 6: Typical SAR data processing, which produces level-1 images, does not include radiometric corrections and significant radiometric bias remains. The radiometric correction is necessary for the pixel values to truly represent the radar backscatter of the reflecting surface and therefore for comparison of SAR images acquired with different sensors, or acquired from the same sensor but at different times, in different modes, or processed by different processors. (*SNAP Help*)

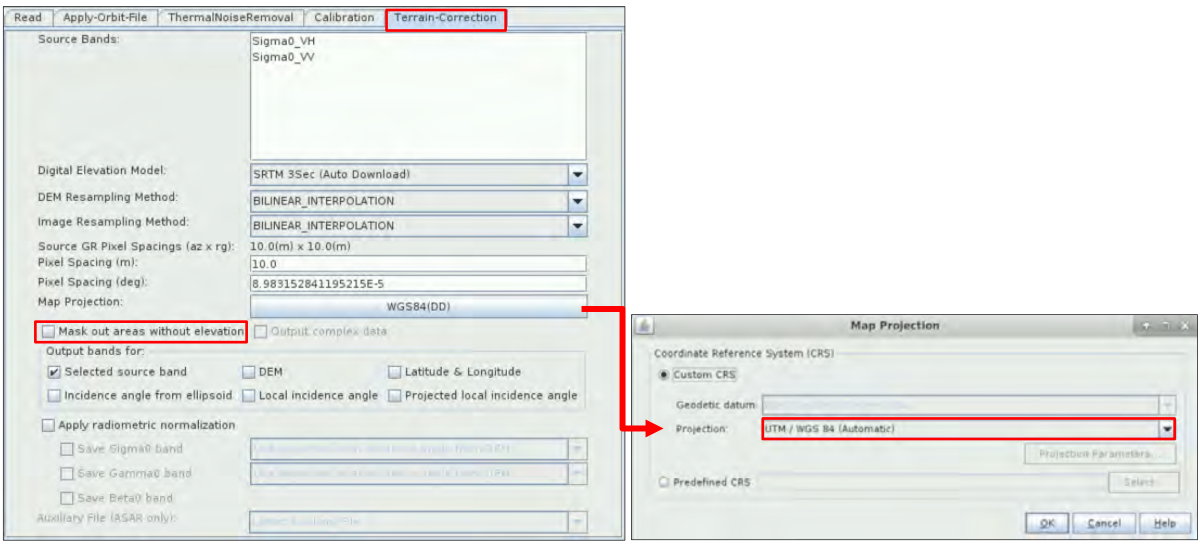


6.3.5 Terrain correction

Our data are still in radar geometry, moreover due to topographical variations of a scene and the tilt of the satellite sensor, the distances can be distorted in the SAR images. Therefore, we will apply terrain correction to compensate for the distortions and reproject the scene to geographic projection (See NOTE 7). To add the operator to our graph, right click and navigate to *Add -> RADAR -> Speckle Filtering -> Speckle-Filter*. In the corresponding tab, make sure you select *UTM / WGS 84 (Automatic)* as Map Projection. Uncheck the option *Mask out areas without elevation* (this option is only for visualization purposes). Connect the *Terrain-Correction* operator with the *Calibration* operator by clicking to the right side of the *Calibration* operator and dragging the red arrow towards the *Terrain-Correction* operator.

NOTE 7: The geometry of topographical distortions in SAR imagery is shown on the right. Here we can see that point **B** with elevation **h** above the ellipsoid is imaged at position **B'** in SAR image, though its real position is **B"**. The offset **Δr** between **B'** and **B"** exhibits the effect of topographic distortions. *(SNAP Help)*

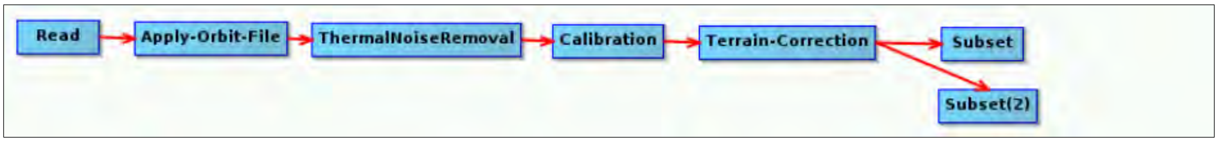




6.3.6 Subset

To investigate the capabilities of the different polarizations to detect rice growth, we will save as independent files the VV and VH polarizations. In this way we will be able to analyse the temporal evolution of the backscatter coefficient for every channel independently. At the same time, we will reduce the original extent of the image. This will reduce the size of the product and processing time.

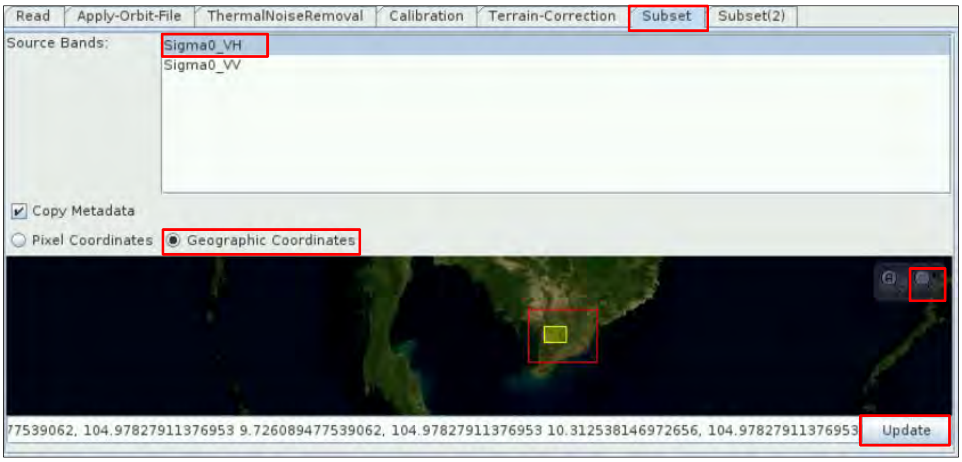
For this, add two *Subset* operators. Right click and navigate to *Add -> Raster -> Geometric -> Subset.* Next, connect the *Subset* and *Subset(2)* operators with the *Terrain-Correction* operator by clicking to the right side of the *Terrain-Correction* operator and dragging the red arrow towards the *Subset* operators.

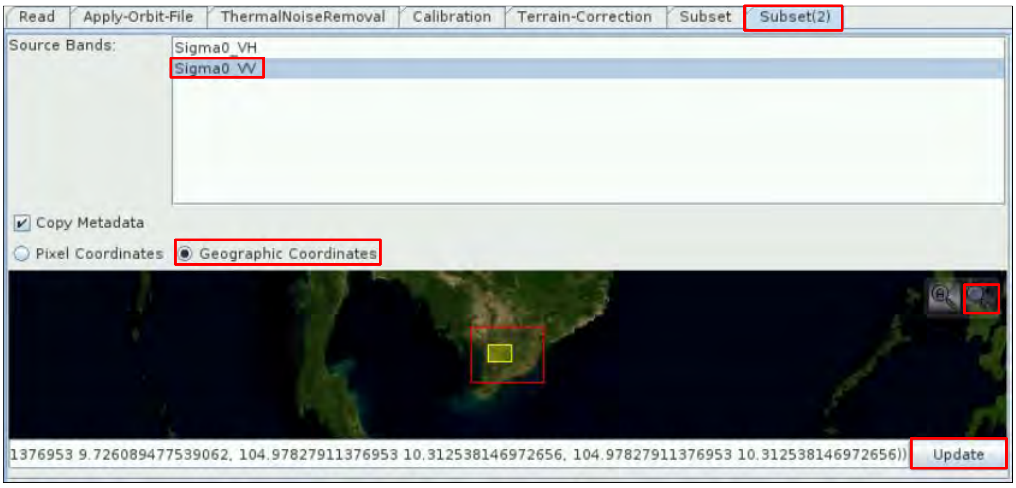


In the first *Subset* tab, select the band ***Sigma0_VH*** choose the option *Geographic Coordinates*. Paste the following Well-Known Text to define to subset area. Then, click *Update* and visualize the area (click on the zoom icon -).

```
POLYGON ((104.97827895096175 10.312537866158335, 105.80329350853457 10.311540351754
259, 105.80185401875113 9.726089296805576, 104.97831787983566 9.727029083669775, 10
4.97827895096175 10.312537866158335))
```

In the second *Subset(2)* tab, select the band ***Sigma0_VV*** and choose the option *Geographic Coordinates*. Paste again the same Well-Known Text to define to subset area. Then, click *Update* and visualize the area by clicking on the zoom icon.

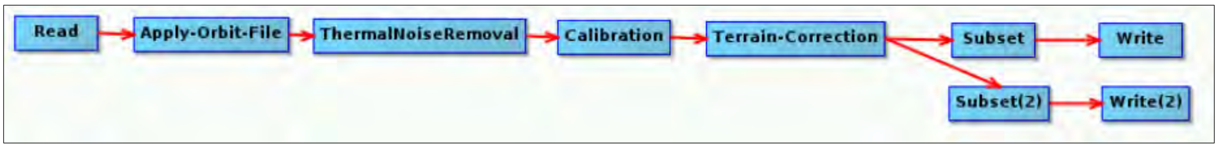


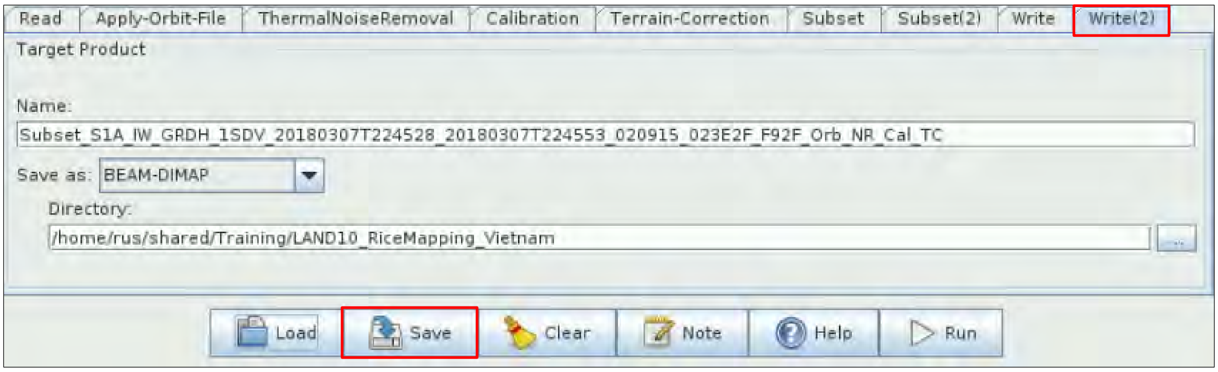


6.3.7 Write

Finally, add the *Write* operators. Right click and navigate to *Add -> Input-Output -> Write*. Add two *Write* operators. Although we will change the output name and output directory in GPT, to avoid confusion set the output directory to the following path.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/*





Once the graph is completed, click on the *Save* icon located on the lower part of the graph builder. Navigate to the following path and save the graph as *S1_Orb_Thm_Cal_TC_sub.xml.* After saving the graph, close the Graph Builder window and SNAP.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/AuxData/*

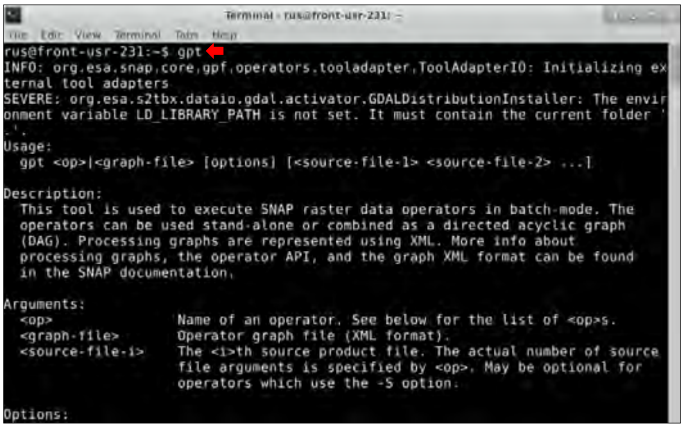
6.3.8 Graph Processing Tool | GPT

In this exercise, we will use the SNAP GPT command line interface (which can be found in the bin folder of the Sentinel Toolbox installation) to process our Sentinel-1 products. This tool is used to execute SNAP raster data operators in batch-mode. The operators can be used stand-alone or combined as a directed acyclic graph (DAG). Processing graphs are represented using XML files. Using the GPT provides a convenient way to use operators in a headless environment or in batch mode (See NOTE 8).

NOTE 8: To run an operator using the GPT, it is necessary to indicate the path to the source product(s), to the target product and to other operator-specific parameters which might be mandatory or specific. As for complex operators the call from the command line can easily become confusing, it is also possible to pass the required settings in form of a xml-encoded graph file. It will then suffice to just pass the graph as parameter to the GPT.

To access GPT, open a *Terminal* window by clicking on its icon - - write the following text and press enter (See NOTE 9).

```
gpt
```



NOTE 9: Note that in the RUS Copernicus Virtual Machines, the gpt command is an environment variable and can be called directly from the terminal. If this is not your case, you will have to set it or specify the path to gpt to call the program.

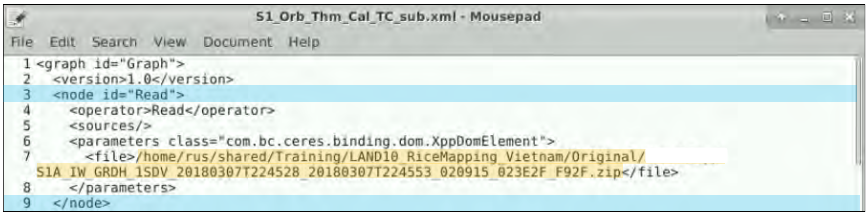
To process our images in batch mode using GPT we need to change the input and output reference to specific input/output files by variables (See NOTE 10). Navigate to the following path, right click on the graph file (called *S1_Orb_Thm_Cal_TC_sub.xml*) and select *Open With -> Open with Mousepad.* Once the *xml* file is opened, click on *View -> Line Numbers*.

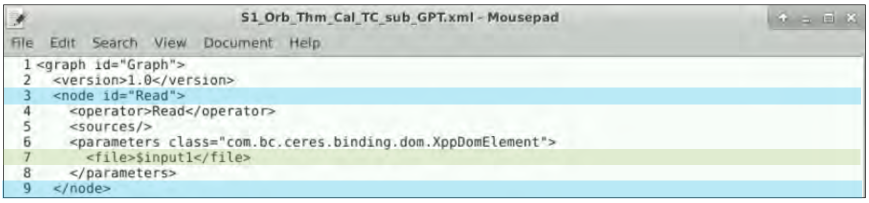
Path: */shared/Training/LAND10_RiceMapping_Vietnam/AuxData/*

NOTE 10: The graph created in the *Graph Builder* tool in SNAP is an xml document that contains the different operators that have been added. The xml document is structured in a way that all the information of a specific operator is specified between the <node> and </node> tags.

In **line 7**, delete only the path to the input image highlighted in orange (do not remove *<file>* and *</file>)* and write *$input1*. Line 7 should look like this (highlighted in green):

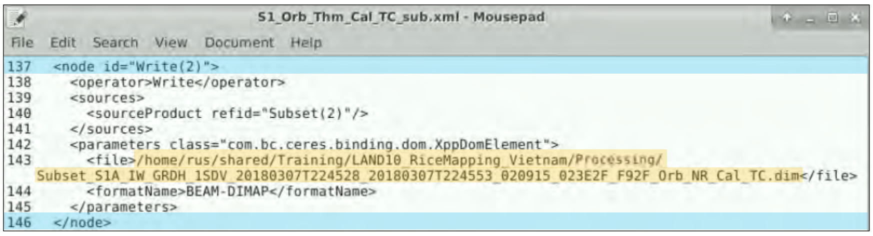
```
<file>$input1</file>
```

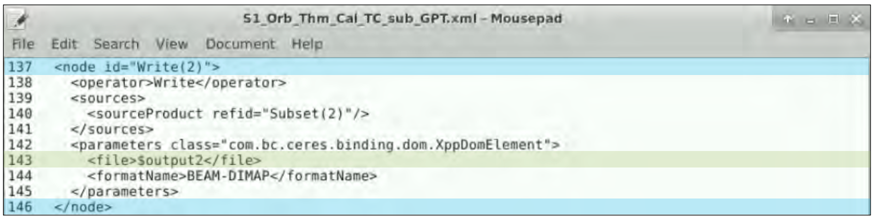




In **line 143**, delete only the path to the output image highlighted in orange (do not remove *<file>* and *</file>*) and write ***$output2***. Line 143 should look like this (highlighted in green):

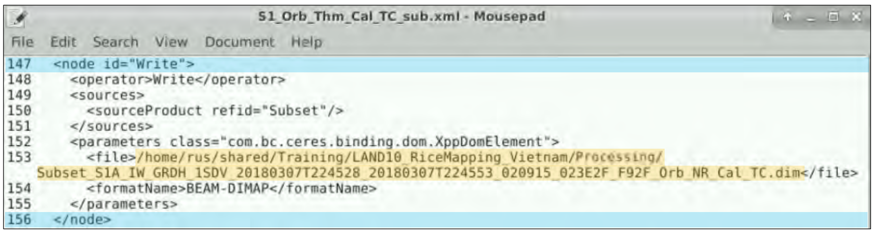
```
<file>$output2</file>
```

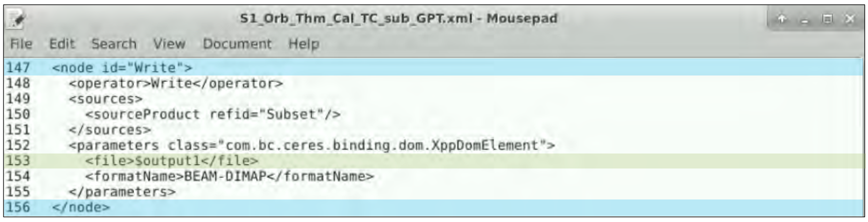




In **line 153**, delete the path to the output image highlighted in orange (do not remove *<file>* and *</file>)* and write ***$output1***. Line 153 should look like this (highlighted in green):

```
<file>$output1</file>
```





Once the input and output variables are defined, save the graph as a new xml file. Go to *File->Save As.* Navigate to the following path and save it as *S1_Orb_Thm_Cal _TC_sub_GPT.xml.*

Path: */shared/Training/LAND10_RiceMapping_Vietnam/AuxData/*

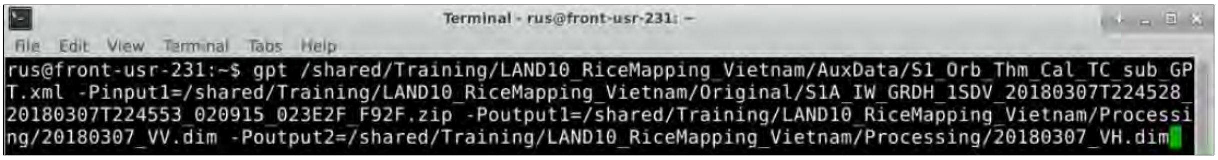
Now, the *$input1*, *$output1* and *$output2* variables allow the graph to be used with different images. However, to run the graph, the value of the variables has to be properly set so that GPT knows which file to use as input and where to save the outputs. When running a graph with variables in GPT, we have to use the *'-P'* option followed by the name of the variables we have created. In an XML graph, all occurrences of ${<name>} will be replaced with <value>.

To run a graph with variables in GPT, the following structure is used:

```
gpt /path/to/graph/ -Pinput1=/path/input/Sentinel/images/ -Poutput1=/path/output1/directory/file.dim -Poutput2=/path/output2/directory/file.dim
```

For example:

```
gpt /shared/Training/LAND10_RiceMapping_Vietnam/AuxData/S1_Orb_Thm_Cal_TC_sub_GPT.xml -Pinput1=/shared/Training/LAND10_RiceMapping_Vietnam/Original/S1A_IW_GRDH_1SDV_20180307T224528_20180307T224553_020915_023E2F_F92F.zip -Poutput1=/shared/Training/LAND10_RiceMapping_Vietnam/Processing/20180307_VH.dim -Poutput2=/shared/Training/LAND10_RiceMapping_Vietnam/Processing/20180307_VV.dim
```



6.3.9 Batch processing

Once the graph is modified with the *$input1*, *$output1* and *$output2* variables, we are ready to process them in batch mode with GPT. For that, when an image is processed, the *$input1*, *$output1* and *$output2* variables have to change with the appropriate name. There are several approaches to do so but, in this exercise, we will use a bash script (See NOTE 11).

> NOTE 11: Bash is a Unix shell and command language written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell. Bash is a command processor that typically runs in a text window where the user types commands that cause actions. Bash can also read and execute commands from a file, called a shell script. Like all Unix shells, it supports filename globbing (wildcard matching), piping, here documents, command substitution, variables, and control structures for condition-testing and iteration. The keywords, syntax and other basic features of the language are all copied from sh. Other features, e.g., history, are copied from csh and ksh. Bash is a POSIX-compliant shell, but with several extensions.

Go to the following path */shared/Training/LAND10_RiceMapping_Vietnam/AuxData/* and click on File -> Create document -> Empty file. Name it *Script_GPT.sh* and click *Create.* See NOTE 12 for some basic information regarding bash.

> NOTE 12: The hash exclamation mark (#!) character sequence is referred to as the Shebang. Following it is the path to the interpreter (or program) that should be used to run (or interpret) the rest of the lines in the text file (for Bash scripts it will be the path to Bash, but there are many other types of scripts and they each have their own interpreter). The shebang must be on the very first line of the file. There must also be no spaces before the # or between the ! and the path to the interpreter. More information on bash scripting on: https://ryanstutorials.net/bash-scripting-tutorial/

The scrip that we will use first defines three variables that are used to set specific paths:

- *Path_S1*: path to folder containing the original Sentinel-1 products (Path_S1)
- *Path_VH*: path to folder where S1 processed images with VH polarization will be saved
- *Path_VV*: path to folder where S1 processed images with VV polarization will be saved

Next, the script defines three other variables:

- *oldEnd*: used to identify original products using the *.zip* pattern
- *VVpol*: used to define the name of the processed images in VV polarization
- *VHpol*: used to define the name of the processed images in VH polarization

Next, a *for* loop starts. All the Sentinel-1 images in the specify directory are identified and a list containing the names is created.

In each iteration (for each element of the list), the sensing time of the product is extracted from the file name and saved in the variable *'n'.* Then, the element of the list is passed in to the *-Pinput1* variable and processed using the predefined graph. The *-Poutput1* variable is defined by adding:

- *Path_VH* → path of the output directory
- *n* → sensing time of the Sentinel-1 product
- *VH_pol* → polarization

For example:

```
                        Path_VH + n + VH_pol

/shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/ + 20180307 + _VH

/shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/20180307_VH.dim
```

The same procedure is used to define the *-Poutput2* and the *-Poutput3* variables. The *date* command provides the starting and finish time of the gpt command.

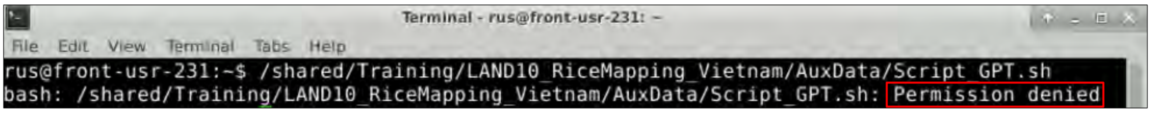
Now that we have a better idea of what the bash script will do, right click on the *Script_GPT.sh* we have created and select *Open with Mousepad.* Copy-paste the following text and remember to save the file after pasting the script (*File -> Save*).

```
#!/bin/bash

# 1- Define path variables
Path_S1=/shared/Training/LAND10_RiceMapping_Vietnam/Original/
Path_VH=/shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/
Path_VV=/shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/

# 2- Define name variables
oldEnd=.zip
VHpol=_VH
VVpol=_VV

# 3- Extract date & run GPT
for i in $(ls -d -1 $Path_S1$S1*.zip)
do
n=${i%.*}
n=${i%T*}
n=${n#"${n%_*}_"}
    date
    gpt  /shared/Training/LAND10_RiceMapping_Vietnam/AuxData/S1_Orb_Thm_Cal_TC_sub_GPT.xml -Pinput1=$i -Poutput1="$Path_VH$n$VHpol" -Poutput2="$Path_VV$n$VVpol"
    date
done
```

Once the script is saved, we can run it. Open a new terminal window, copy-paste the following command and press enter to run it. If you get a *Permission denied* error, see NOTE 13. You can monitor the usage of RAM memory by opening the *Task Manager* (Applications -> System -> Task Manager). This process may take some time depending on the number of images to process, the processing steps applied and your IT environment.

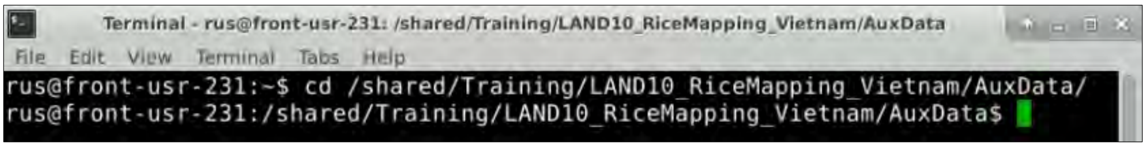
```
/shared/Training/LAND10_RiceMapping_Vietnam/AuxData/Script_GPT.sh
```

NOTE 13: When running the *Script_GPT.sh* bash script, it can happen that the following error appears in Terminal: *Permission denied*



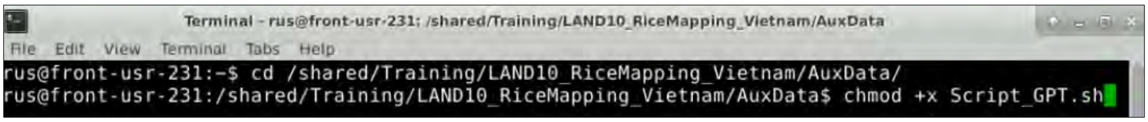
It is cause by the lack of an execute permission bit for the file you want to run. To solve it, we just need specify that the file can be executed. For that, access the folder where the file is stored. Copy paste the following code in Terminal and run it (Press Enter).

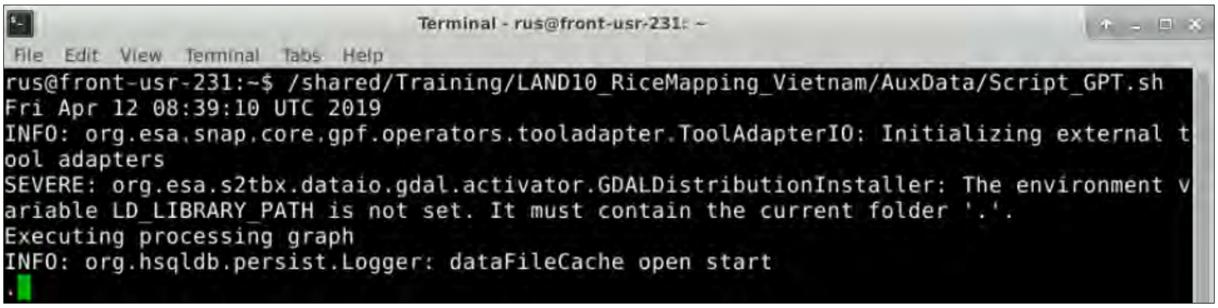
```
cd /shared/Training/LAND10_RiceMapping_Vietnam/AuxData/
```

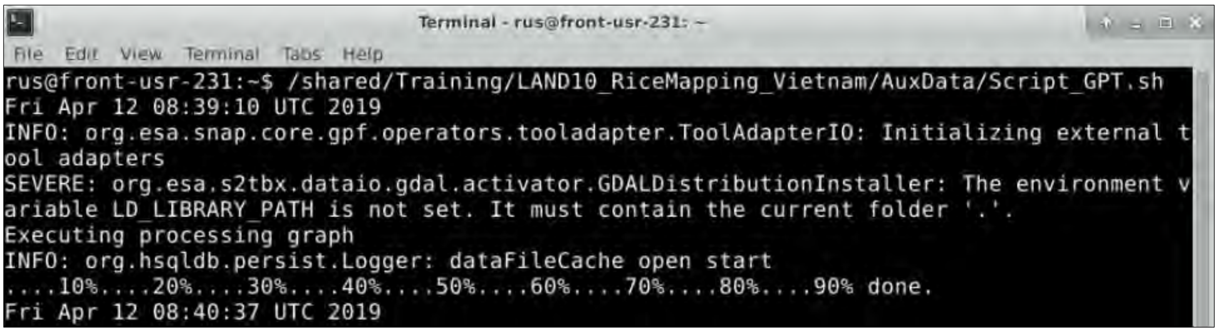


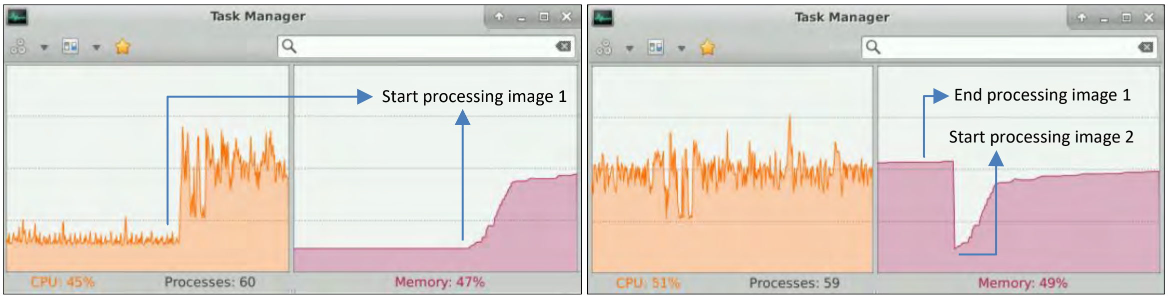
Next, copy-paste the following command to make the file executable and run it by pressing enter.

```
chmod +x Script_GPT.sh
```





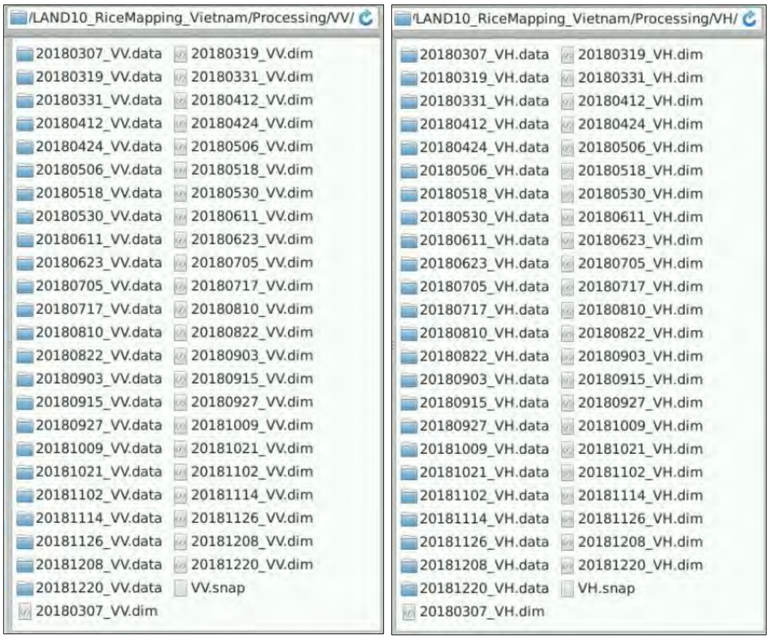




Once the script is finished and all the images have been processed, you will find them in the paths specified in the bash script. Navigate to the following paths and check the files have been created.

Path 1 → /shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/

Path 2 → /shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/



6.3.10 Stack

After the images have been processed, we will stack them together according to their polarization to allow further processing.

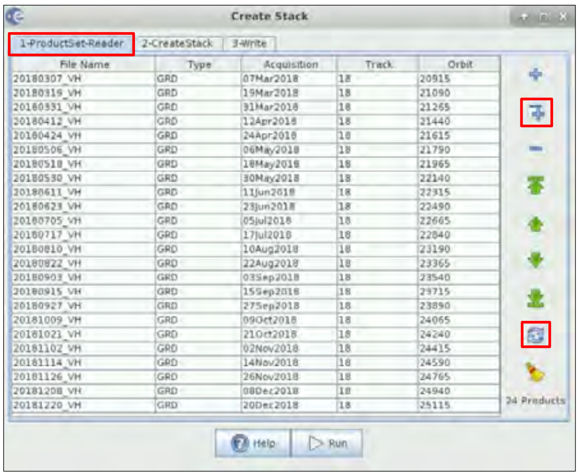
To prevent errors when creating the stack, images have to be opened in chronological order. For convenience SNAP session files have been created in advanced and are located in the following paths (See NOTE 14). In SNAP go to File -> Session -> Open Session. Navigate to Path_1 and open the *VH_images.snap* file. Next, go to *RADAR -> Corregistration -> Stack Tools -> Create Stack.*

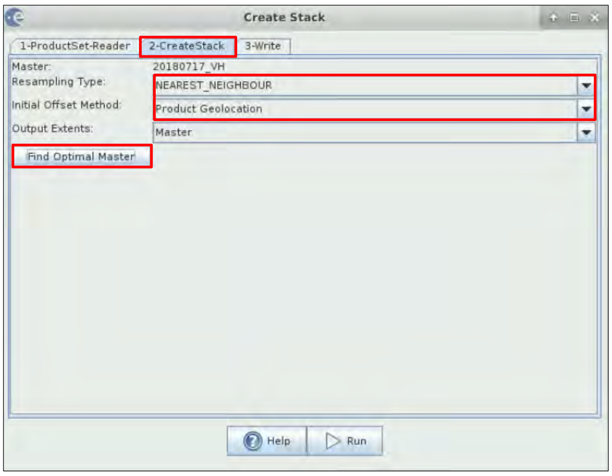
> NOTE 14: Note that opening the SNAP session file will only work if Sentinel-1 images have been processed and stored in the appropriate path as explained in previous steps. If you run batch processing without errors, you should not face any issue. You can also manually load images in chronological order.

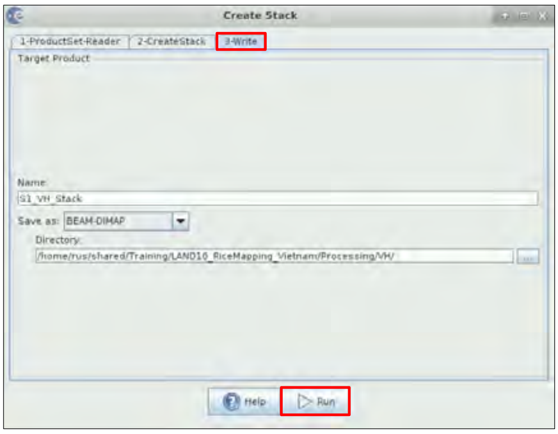
Path_1: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*

Path_2: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/*

In the *1-ProductSet-Reader* click on the *Add Opened* icon - - and click on the *Refresh* - - icon to update the metadata information. In the *2-CreateStack* tab, select *NEAREST_NEIGHBOUR* resampling method, set the initial offset method to *Product Geolocation* and click on the *Find Optimal Master* button. In the *3-Write* tab, change the name to *S1_VH_Stack* and make sure to set the output directory to the *Path_1*. Click Run.







Once finished, repeat the same procedure for the VV polarization images and save them with the appropriate names and in the appropriate directory: *S1_VV_Stack* and *Path_2* respectively.

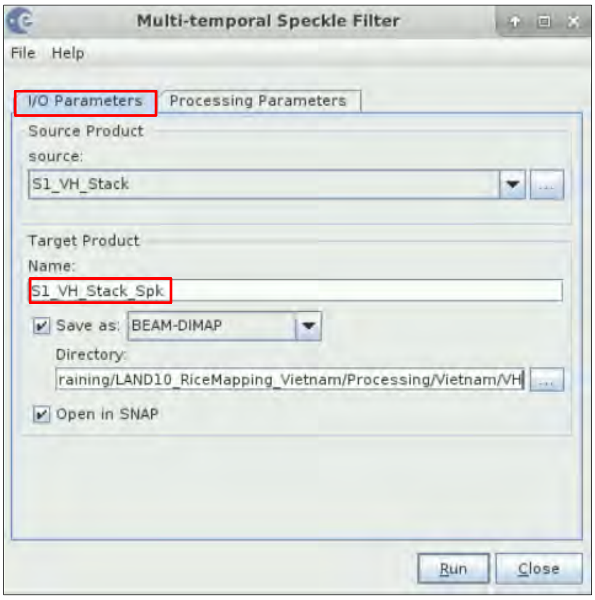
6.3.11 Multi-temporal speckle filtering

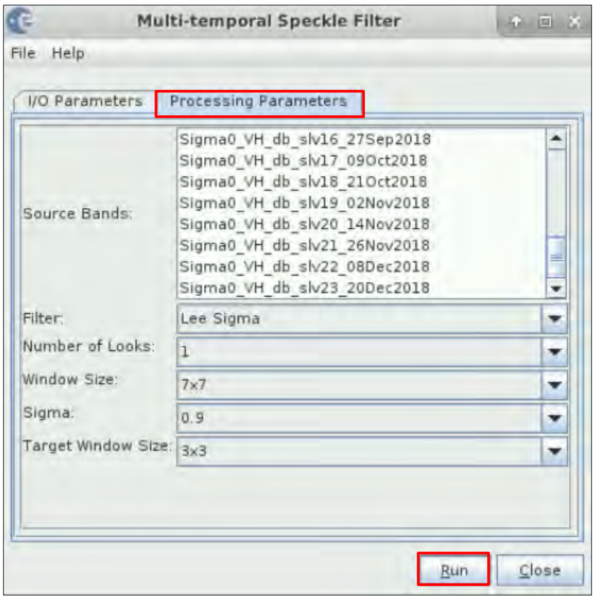
Once the two different stacks have been created, we will reduce the effect of speckle noise in each of them. SAR images have inherent salt and pepper like texturing called speckles that degrade the quality of the image and make interpretation of features more difficult (See NOTE 15). To reduce the speckle effect, we will take advantage of the temporal dataset we are working with and apply a multi-temporal speckle filter. For this exercise, the default filter used in SNAP (Lee Sigma) will be used.

> NOTE 15: Speckle is caused by random constructive and destructive interference of the de-phased but coherent return waves scattered by the elementary scatters within each resolution cell. Speckle noise reduction can be applied by either spatial filtering or multilook processing. Multi-temporal filtering is one of the commonly used speckle noise reduction techniques. (*SNAP Help*)

Go to *RADAR -> Speckle Filtering -> Multi-temporal speckle filter.* First we will reduce the speckle for the *S1_VH_stack.dim* file that was created in the previous step. In the *I/O Parameters* tab click on the icon go to *Path_1* and select the file. Make sure the output name is set to *S1_VH_Stack_Spk* and the output directory to the following path. In the *Processing Parameters* tab leave the default values.

Path_1: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*





Once finished, repeat the same procedure for the *S1_VV_Stack.dim* file. Make sure to save it with the appropriate name and in the appropriate directory: *S1_VV_Stack_Spk* and *Path_2* respectively. Once finished, close all the products in SNAP except for the two stacked files.

Path_2: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/*

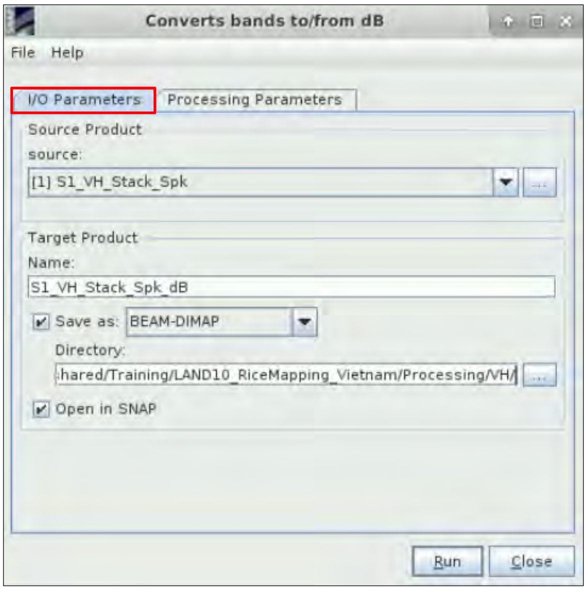
6.3.12 Decibel transformation

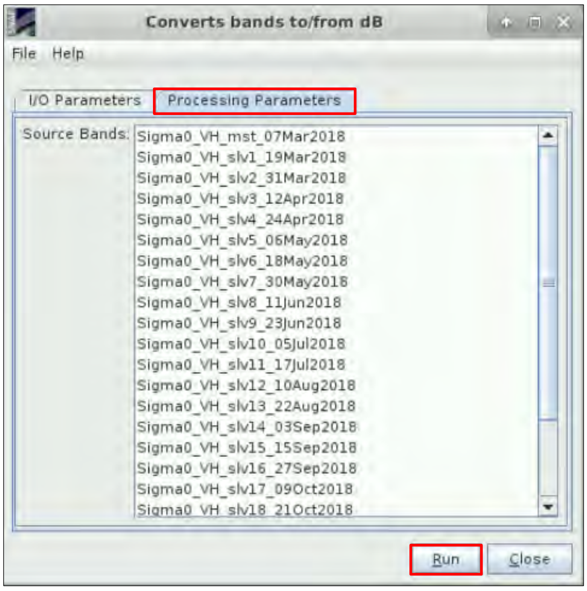
Due to the high dynamic range of SAR imagery, the decibel transformation is used to improve visualization and data analysis. The transformation will stretch the RADAR backscatter over a more usable range which has nearly a gaussian distribution. The RADAR backscatter coefficient (σ^0) is transformed into the decibel scale using the following equation:

$$\sigma^0\ (dB) = 10 * \log 10\sigma^0$$

Go to *Raster -> Data Conversion -> Convert bands to/from dB*. In the *I/O Parameters* tab, make sure to select the *S1_VH_Stack_Spk* as input and set the output directory to the following path. Leave the default settings in the *Processing Parameters* tab.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*





Once finished, repeat the same procedure for the *S1_VV_Stack_Spk* file and save the output in the following path:

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/*

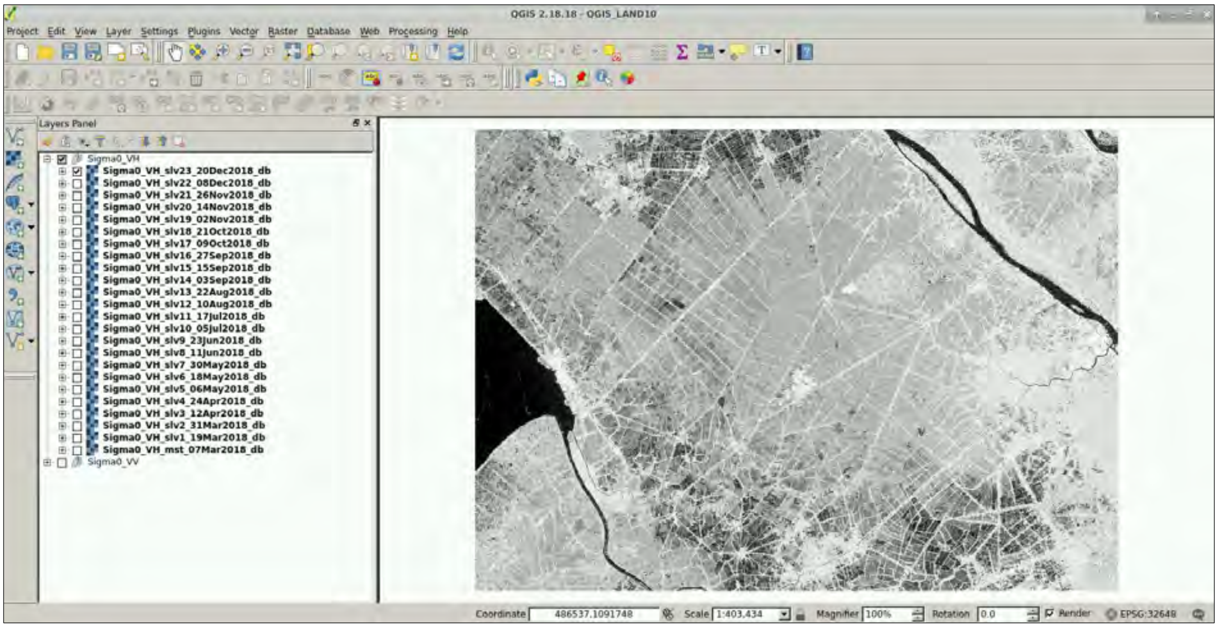
6.4 QGIS

6.4.1 Sigma Naught evolution

Once the conversion to the decibel scale is finished, close SNAP and open QGIS (*Applications -> Processing -> QGIS Desktop).* For convenience, a session containing the images we will use to analyse the temporal evolution of the backscatter coefficient has been created in advance. Go to *Project -> Open*, navigate to the following path and open the *QGIS_LAND10.qgs* file (See NOTE 16).

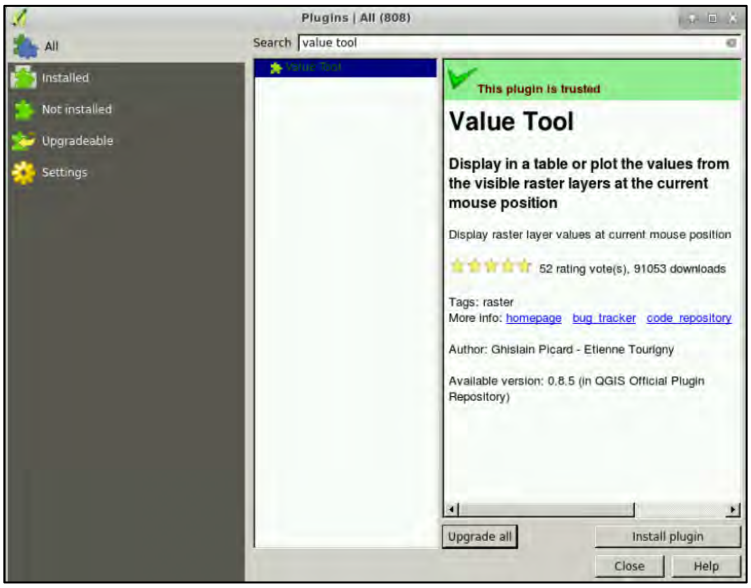
Path: */shared/Training/LAND10_RiceMapping_Vietnam/AuxData/*

NOTE 16: The QGIS session file that is provided in this exercise will only work if you have saved the processed Sentinel-1 files with the same name and in the same path as specified in this tutorial. If not, you can always load the products to QGIS manually.



To analyse the evolution of the backscatter coefficient we will use the *Value Tool* (See NOTE 17), a specific plugin that allows to plot pixel values in different layers with the mouse position as reference.

NOTE 17: If you do not have the *'Value Tool'* plugin install, click on the menu *Plugins -> Manage and install plugins.* Select the *'All'* section on the right-side panel*, write 'Value Tool'* on the search box, select the *'Value Tool'* and click install.

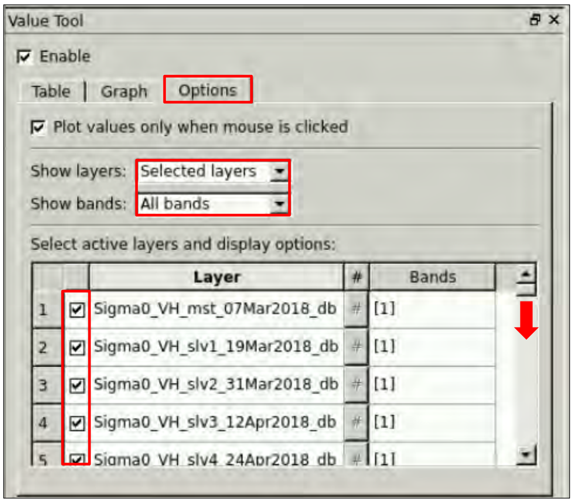


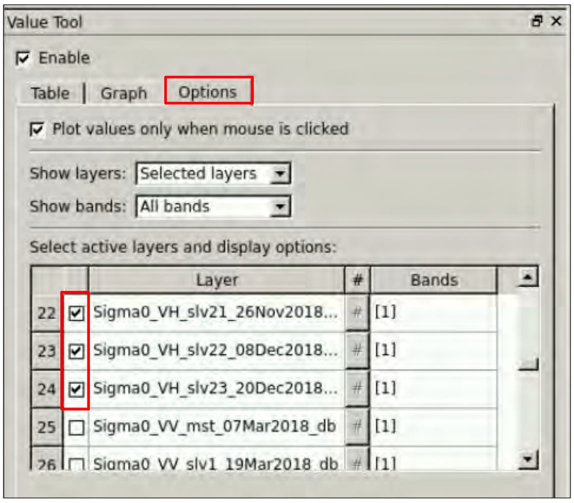
Once installed, the tool will appear in the lower left corner (if not visible, right click on the Toolbar and select *Value Tool).* Activate the tool by clicking on *Enable*, go to the *Options* tab and click on the option *Plot values only when mouse is clicked.* Set the following parameters:

Show layers: *Selected layers*

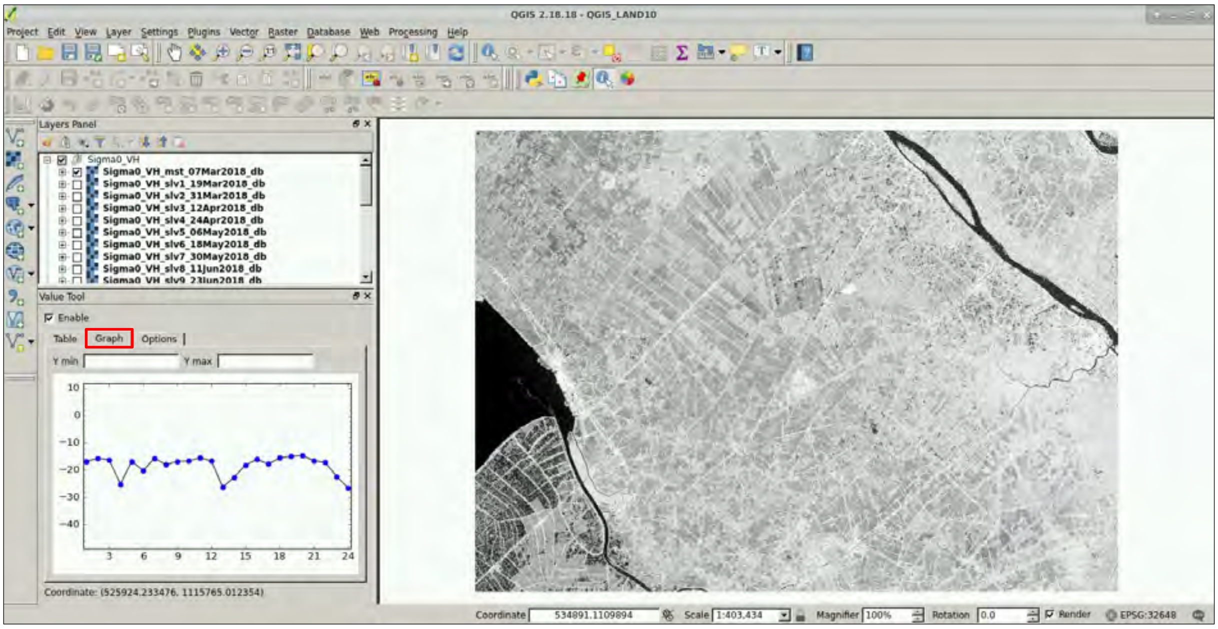
Show bands: *All bands*

In the list of layers of the *Value Tool*, select all the VH (or VV) images (24 in total) to analyze the temporal profile.



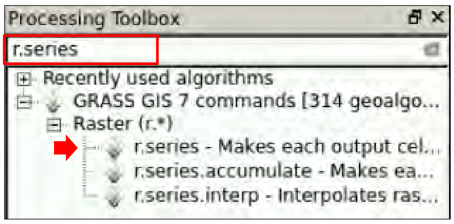


Once all the parameters are set, click on the *Graph* tab*.* Zoom in a specific area of the image, place your mouse over a field and click. The graph will then show the Sigma Naught (σ0) evolution in all the Sentinel-1 VH (or VV) images we have previously processed.

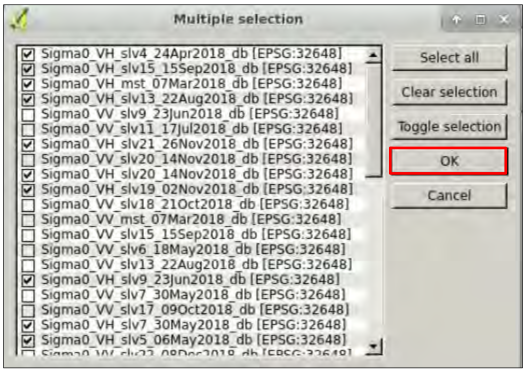


6.4.2 QGIS Processing

In the last step we will take advantage of the large differences of backscatter produced by the different growing stages of rice to highlight the fields where this crop is been cultivated. Go to *Processing -> Toolbox.* In the search box, write *r.series* and open the tool.

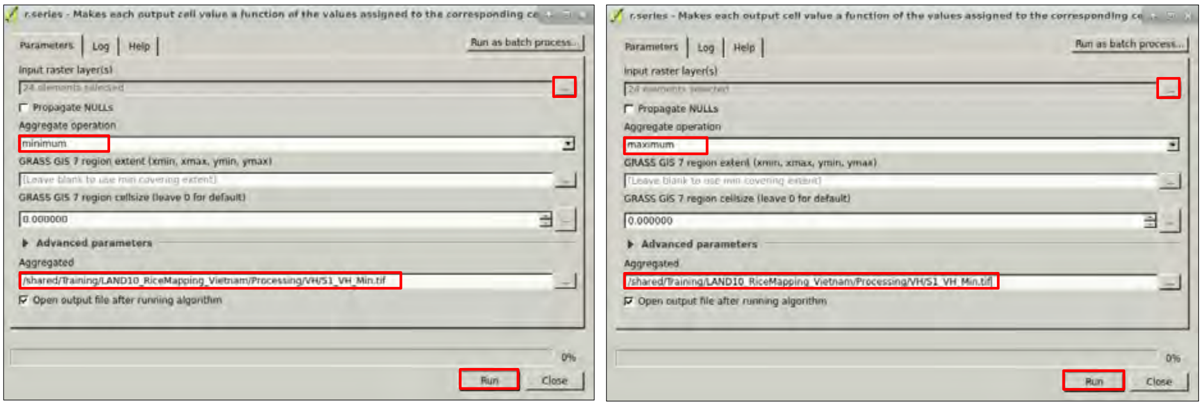


In the *Input raster layer(s),* click on the  icon, select all the VH bands and click OK.



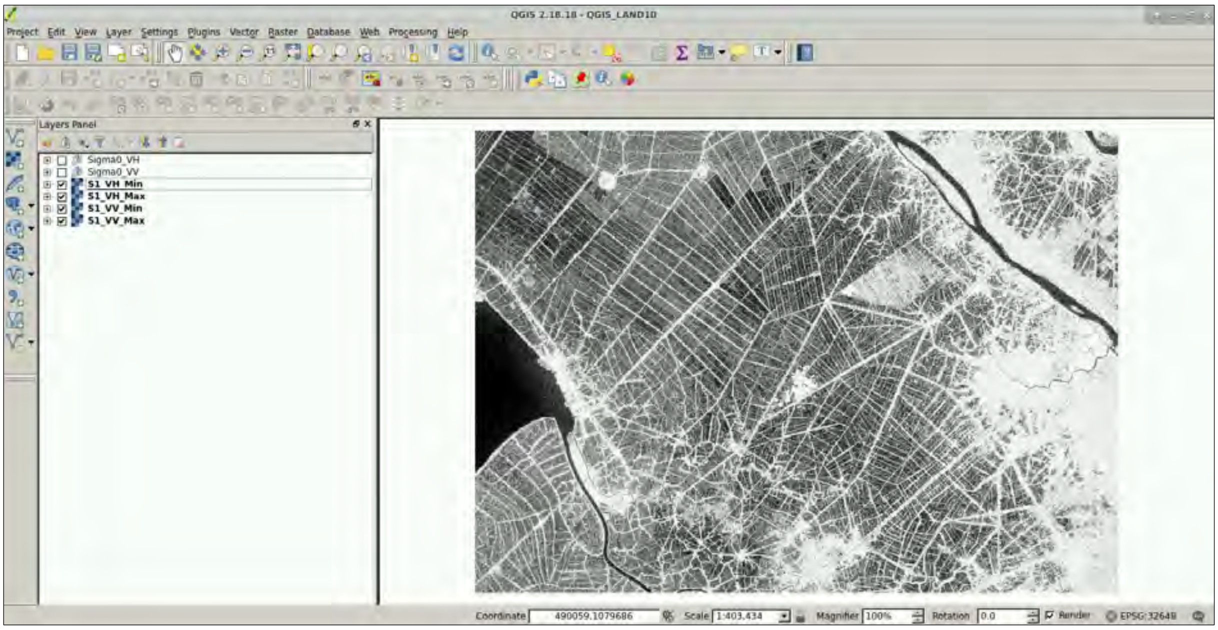
In the *Aggregate operation* tab, select *minimum*. Leave the remaining parameters as default and save the output as *S1_VH_Min* in the following path. Then, click *Run*. Once finished, change the *Aggregate operation* to *maximum* and save the output as *S1_VH_Max* in the same directory as before.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*



Once the minimum and maximum value for each pixel has been derived for the Sentinel-1 VH images, do the same for the VV layers. Change the *input raster layer(s)* and derive the same statistics. Name them *S1_VV_Min* and *S1_VV_Max* and save them in the following path:

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/*

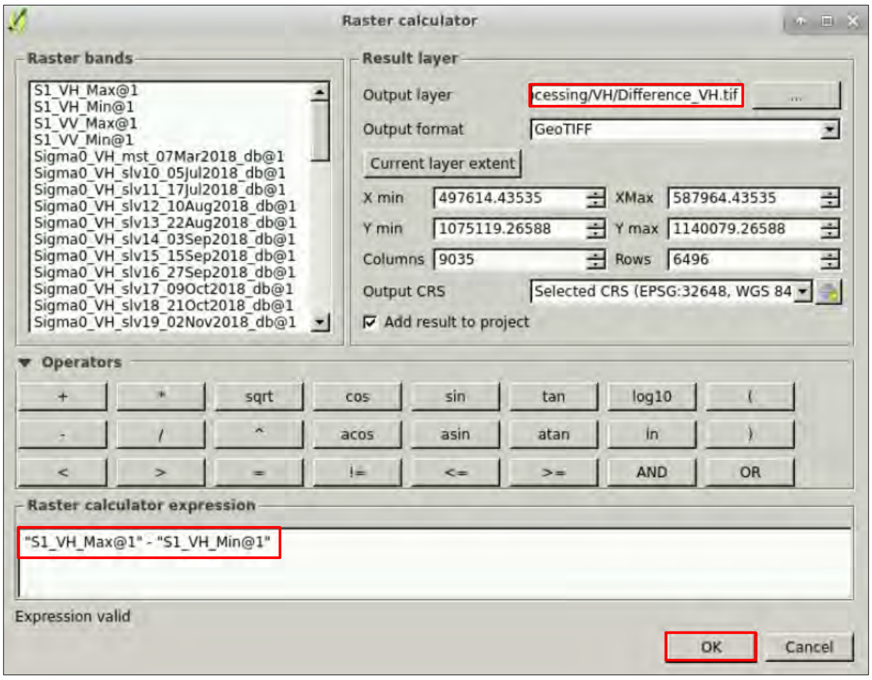


Next, we will derive the difference between the minimum and maximum value for both polarizations. Go to *Raster -> Raster Calculator.* Set the output name as *Difference_VH* and save it in the following path.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*

Leave the remaining parameters as default and copy-paste the following expression under the *Raster calculator expression*. Then, click Ok.

"S1_VH_Max@1" - "S1_VH_Min@1"

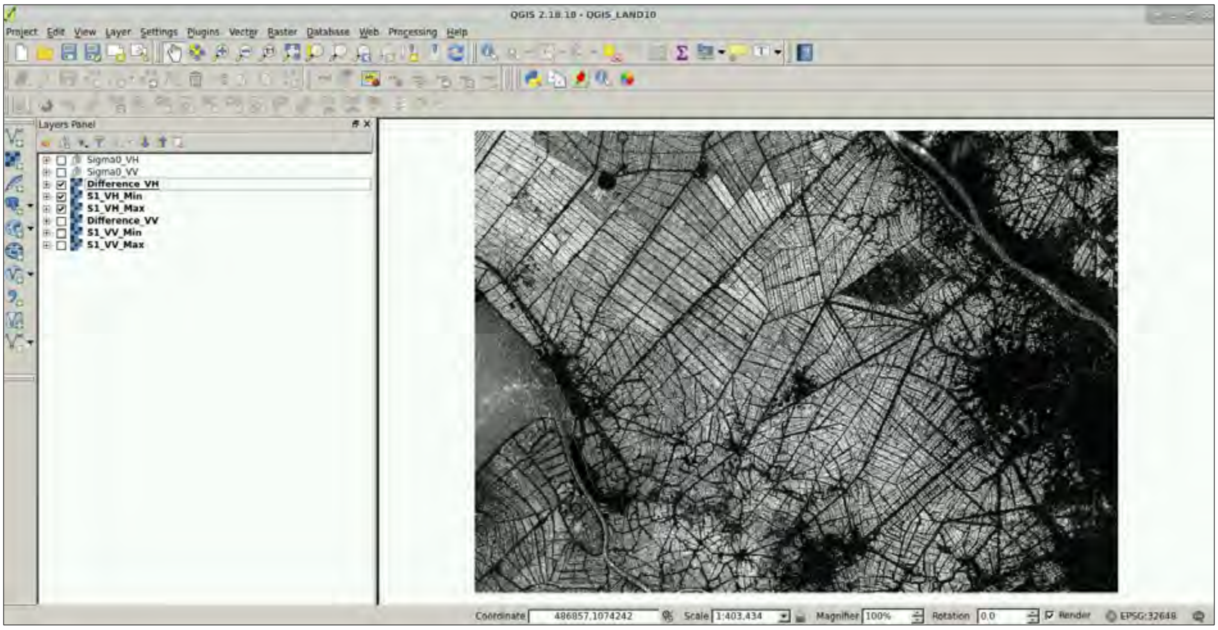


Once finished, repeat the same procedure to derive the difference between the *S1_VV_Max* and *S1_VV_Min* layers. Set the output name as *Difference_VV* and save it in the following path. Leave the remaining parameters as default and copy-paste the following expression. Then, click OK.

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VV/*

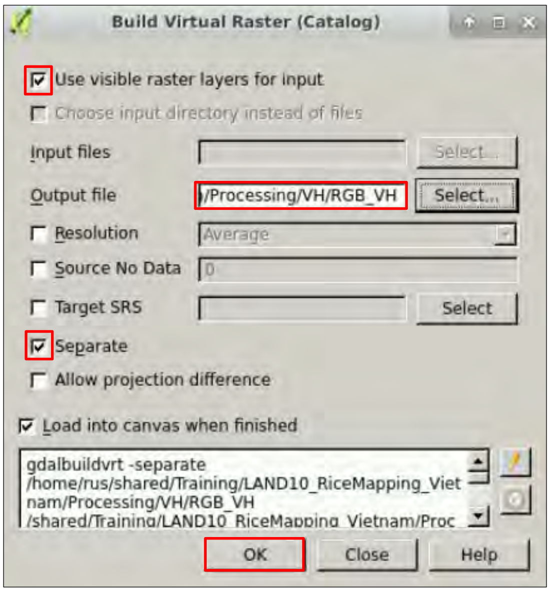
"S1_VV_Max@1" - "S1_VV_Min@1"

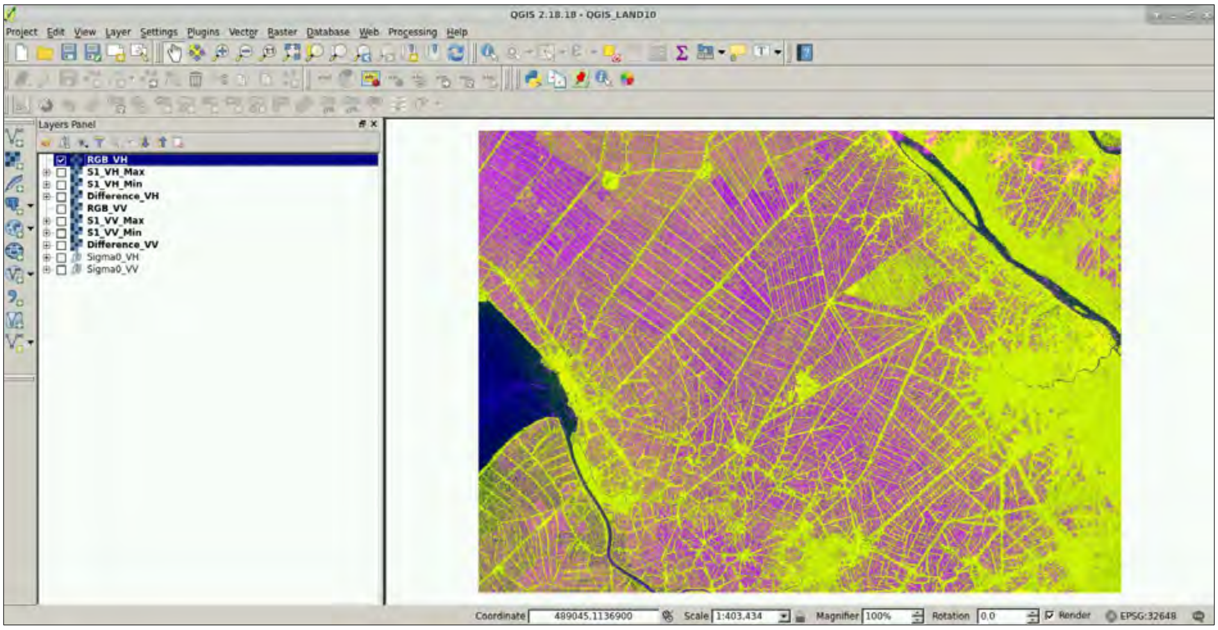
Finally, we will create a false-colour RGB composition to better visualize the crop patterns. In the *Layers Panel,* select only the *Difference_VH, S1_VH_Max* and *S1_VH_Min* layers. Next, go to *Raster -> Miscellaneous -> Build Virtual Raster*



Select the option *Use visible raster layers for input*, save the output file as *RGB_VH* in the following path and check the option *Separate.* Then, click Ok. Repeat the same procedure for the *Difference_VV, S1_VV_Max* and *S1_VV_Min* layers and save it as *RGB_VV* in the corresponding folder. (You may have different color patterns depending on how the layers have been assigned to the RGB channels).

Path: */shared/Training/LAND10_RiceMapping_Vietnam/Processing/VH/*





THANK YOU FOR FOLLOWING THE EXERCISE!

7 Further reading and resources

Sentinel-1 User Guide

https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar

Sentinel-1 Technical Guide

https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-1-sar

SNAP GPT Guide

https://senbox.atlassian.net/wiki/spaces/SNAP/overview

Bash and Linux Tutorial

https://ryanstutorials.net/bash-scripting-tutorial/

FOLLOW US!!!

@RUS-Copernicus

RUS-Copernicus

RUS-Copernicus

RUS Copernicus Training

RUS-Copernicus website

RUS-Copernicus Training website