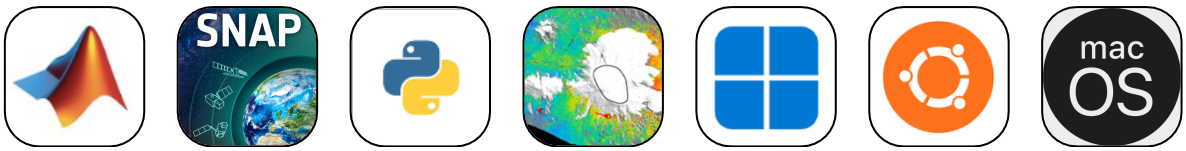
PHASE

Persistent scatterer
Highly
Automated
Suite
for Environmental monitoring

MANUAL – v2.0

REQUIRED SOFTWARE:
SNAP
STAMPS
MATLAB



Roberto Monti, Lorenzo Rossi
Politecnico di Milano

INTRODUCTION

The aim of these MATLAB software suite is to provide tools that can be used to automatize the entire processing of InSAR time series for the PSI analysis, from the download of the images to the creation of a table containing the displacement time series of each Persistent Scatterer (PS). To do so, it is based on the well-known and widely used *SNAP* and *StaMPS* software. Despite their large spread in the InSAR community, there is no existence of an automatized procedure that reduces as much as possible the required user actions, without decreasing the level of control on the processing parameters. For these reasons, and to reduce the waste of time too, two MATLAB applications have been developed.
The first one is based on *SNAP* (through *snap2stamps*) for the pre-processing and can be easily used under both Windows and Linux environment. In order to increase computational performances, it is recommended to execute it in the default operative system of the PC (not in a virtual machine). The second one, instead, relies on the *StaMPS* PS analysis and must be used in a Linux environment due to *StaMPS* compatibility requirements.
The proposed procedure has been developed to work with both Sentinel-1 and Cosmo-SkyMed SAR images; the choice among the two can be easily done by means of a toggle switch.
In the following there will be presented deeply descriptions of the two proposed codes in addition to all the preliminary operations that need to be done before starting the processing.

The provided files, alongside this manual, are two folders, named PHASE_pythonX, where X refers to a different python version (2.7 or 3.x). The code is the same for both cases, just adapted to the python version. The user is invited to download the one compatible with his/her environment. Each folder contains:

- PHASE_Preprocessing.mlapp
- PHASE_Preprocessing folder:
    - PHASE_StaMPS.mlapp
    - snap2stamps folder
- PHASE_logo.png

Finally, I just want to say thank you to Jose Manuel Delgado Blasco and Dr. Michael Foumelis for the development of the *snap2stamps* tool and Prof. Andy Hooper for the development of *StaMPS*.

When using this software suite, please refer to:

Monti, R., & Rossi, L. (2025). PHASE: a Matlab-based software for the DInSAR PS processing. *Geodesy and Cartography*, *51*(2), 88–99. https://doi.org/10.3846/gac.2025.21995

Last update: 05 June 2025
Version 2.0

1. AUTOMATED SAR IMAGES DOWNLOAD & SNAP2STAMPS PRE-PROCESSING

1.1 Preliminary steps

The first step is to install SNAP software from the European Space Agency website, as *snap2stamps*[1] relies on its libraries. Then, some parameters should be reviewed accordingly to what is suggested in the *snap2stamps* user manual and distributed with the code. The suggestions can be found on the GitHub repository of the PHASE software suite. Also, some not standard python modules may be required to be installed too.
The entire procedure has been tested on SNAP 10.x, python 2.7, python 3.11, Ubuntu 20.04, MATLAB 2023b, Windows 10 and macOS Sequoia (15.1).

☼ SUGGESTION To take advantage of the full computational power of your machine, it is suggested to run this first application in the native OS. Remember that PHASE_Preprocessing.mlapp is fully compatible with Windows, macOS and Linux.

For users working in a Linux environment, an additional preliminary step is required for the correct functioning of the entire code. It is, indeed, required to have installed the software *xterm*. If it is not already done, it can be easily done by opening the terminal and typing:

sudo apt-get install xterm

This will just be used to open terminal windows during the process when user input is required or to monitor the processing progress.

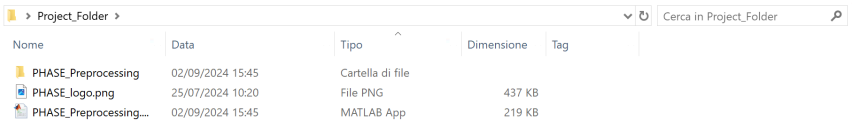
1.2 Required folder structure

Two versions of the software are provided: one compatible with python 2.7, and one with python 3.x. The user has to take the one suitable for the python version installed in his/her machine. There are no requirements for a predefined folder/subfolder structure. On the contrary, the user is free to copy and paste the *PHASE_Preprocessing.mlapp* (MATLAB application), the logo image and the PHASE_Preprocessing folder wherever he/she wants. All the given files do not need to be moved from their given position. So, to recap the procedure:

1. Download the *PHASE_MANUAL* and the *PHASE_python2* or *PHASE_python3* folder based on your python version.

   ☼ SUGGESTION To check the python version and naming accessible through your MATLAB software just type *pyenv* in the MATLAB command window.

2. Move or copy the downloaded folder (*PHASE_pythonX*) in your project folder, anywhere on your computer.



3. Execute the *PHASE_Preprocessing.mlapp* MATLAB application.

1.3 Provided snap2stamps files

Inside the snap2stamps folder, there are two subfolders: bin and graphs. The bin one contains all the python scripts required for each pre-processing step and they do **not** require any modification. Inside the graphs one, instead, there are the .xml files, and again **no** modifications are required.
Contrary to what it is distributed by the original developers, the scripts now have a prefix in their name to distinguish between the two satellite platforms (SEN for Sentinel-1, CSK for Cosmo-SkyMed). All the scripts, starting from the distributed version, have been updated or created as new, with the aim to match all the possible processing configurations.
A brief description of each .xml code is provided below.

- SEN_master_split_applyorbit: master split and orbit correction.
- SEN_slave_split_applyorbit: slave split and orbit correction.
- SEN_coreg_ifg_computation_subset_SINGLE_BURST: slave subset, coregistration and interferogram, for a single burst; therefore, it does not contain the ESD operator.
- SEN_coreg_ifg_computation_subset_MULTI_BURST: slave subset, coregistration and interferogram for multiple bursts; it contains the Enhance Spectral Diversity (ESD) operator.
- SEN_export: slave export in a format compatible with StaMPS.
- SEN_average_intensity: pixel's average intensity over the chosen AOI (it is automatically created by the corresponding .py script at first execution).
- SEN_slave_terrain_correction: slave terrain corrected local incidence angle (LIA) and coherence (COH) images.
- CSK_AOI_subset: SAR image subset.
- CSK_coreg_ifg_computation: slave coregistration and interferogram computation.
- CSK_export: slave export in a format compatible with StaMPS.
- CSK_average_intensity: pixel's average intensity over the chosen AOI (it is automatically created by the corresponding .py script at first execution).
- CSK_slave_terrain_correction: slave terrain corrected local incidence angle (LIA) and coherence (COH) images.

The .xml for the creation of the slaves' average amplitude image for both SEN and CSK cases is automatically created by the corresponding python script once it is reached that point of the processing.
Information about External DEM creation is provided in Appendix A.

⚠ REQUIREMENT It is mandatory to have the following SNAP plugins installed for the correct functioning of the entire workflow: Microwave Toolbox Kit Module, Optical Toolbox Kit Module, SMOS-Box Kit Module, Radarsat Polarimetric Toolkit Module, ESA SNAPPY, EOMTBX.

1.4 Description of the MATLAB application

The MATLAB application is aimed at providing a simple GUI by means of which set all the required variables for the processing, as well as checking the progress of the computation. It has been tested on MATLAB 2023b running on both Windows and Linux environments. The choice of the operating system has to be made from the proper flag in the first panel. The operations performed by the code are summarized in the following list.

1) The user is required to define all the variable parameters needed for both processing and path description. The first step is to move the switch to the satellite constellation which images will be used. Several panels are created in order to make clearer the input parameters and what they refer to.

    a. Operative system: contains the variables identifying the python environment.
    b. Global Variables: contains information regarding the download of the SAR images, the time span of the analysis, the type of orbit, the first and last month.

        ⚠ WARNING The images download flag works only with the python script previously downloaded from the *alaska.edu* portal. They used to provide this script in python 2.7 up to 2024. Therefore, the new downloads work only with PHASE_python3, while PHASE_python2 requires older scripts (likely from previous works).

    c. Master Processing: contains all the parameters required for the correct definition of the master configuration file, its date, as well as the boundaries of the AOI.
    d. Slaves Processing: contains all the parameters required for the correct definition of the slaves' configuration file. There is the possibility to use SNAP built-in DEMs or an external one. The *slaves removal* flag is intended to declare if to delete or not the slave images at the end of the pre-processing, to save disk space.
    e. Coherence and LIA: contains a thick box to perform or not this step and the EPSG code for the images projection. The EPSG code is needed also for the slaves mean amplitude image. Here you can find the code for each region: https://epsg.io.
    f. Computational Resources: contains the global parameters and path for the correct implementation of the InSAR preprocessing steps.

        ☼ SUGGESTION The path to the SNAP gpt/bin folder varies depending on your OS. For **Windows**, it usually is: *C:\Program Files\snap\bin\gpt*. For **macOS**, it usually is: */Applications/snap/bin/gpt*. For **Linux**, it usually is: */usr/local/snap/bin/gpt*.

    g. Save/Load: allows the user to save the configured variables and/or load them from a previous saving.
    h. Run: contains a button to start the execution of the preprocessing steps.

Note that **ALL** the variables must be configured as required for input by the *PHASE_Preprocessing.mlapp* application. Read the comments for each variable in the panels to better understand their meaning. For the latitude and longitude values, just **three decimal** digits are required.

2) All the selected images are then downloaded in the slaves folder.
This is implemented to work only for Sentinel-1 with the Alaska SAR satellite facility website; from this website (https://search.asf.alaska.edu) there can be selected all the images of interest and then a python file containing all their filenames can be downloaded. This is – at the moment – the only method to download SAR images directly from within the application. Note that this requires an active registered account. It is common that at the first usage of the application username and password are requested, that's why an external command window will automatically be opened. In case this step is not necessary, just untick the corresponding box.
On the other hand, Cosmo-SkyMed SAR images need to be manually downloaded from the Italian Space Agency portal.

3) Then, the master image, that has been previously defined (hint: for long time series it is suggested that the chosen one corresponds to a late summer period with clear sky) is automatically moved in the master folder. It is then created the required structure (folder/subfolder) in which the master image is processed. The processing consists in the image splitting and orbit correction with precise ones. The governing parameters are the ones contained in the configuration file of the *snap2stamps* code: polarization, swath, initial burst, final burst, and area of interest. Only a single swath processing is supported at the moment. The execution part is composed of two subsequent steps:
- creation of a *.conf* file with all the parameters in the correct syntax;
- execution of the steps through a batch or bash file (for Windows or Linux, respectively) in the system command prompt.

4) It consists of the common/standard *snap2stamps* pre-processing, plus the average scene intensity, the coherence and local incidence angle estimation. At first, the configuration file of the project is automatically created with the defined information and paths. Then the six steps are:

- slaves preparation (step 1);
- slaves splitting and orbit correction (step 2);
- coregistration and interferogram formation (step 3);
- StaMPS [2] export (step 4);
- average scene intensity (step 5);
- LIA and coherence images computation (step 6).

Similarly as before, everything is executed via a batch (or bash) file and an external command window is opened to better monitor the processing progresses as it may take long.

In case you want to reprocess just a step you can do it by modifying the initial and final step variables in the Slaves Processing panel.
Finally, the folder structure for the *StaMPS* processing is created; for convenience the *StaMPS* processing folder is named with the type of orbit and the initial and final month of the analysis. The

PHASE_StaMPS.mlapp application has **not** to be moved from the PHASE_Preprocessing folder because it will automatically be repositioned in correct position by the given software.

⚠ REQUIREMENT in case the user has **MANUALLY DOWNLOADED ALL THE SAR IMAGES**, they have to be all placed inside the slaves folder (path: PHASE_Preprocessing/slaves). they must be .zip files for sentinel-1 data and .h5 files for COSMO-SkyMed data.

⚠ REQUIREMENT always select all configurable parameters by clicking on the corresponding drop-down menu; do it also if the interface is already showing your desired value.

⚠ WARNING if during the master processing something goes wrong and during the second execution it is not automatically solved, please **MANUALLY MOVE THE MASTER IMAGE FILE** (.zip or .h5) back to the slaves folder and delete the master folder.

Finally, in the following, there are reported the SNAP operators used in the main processing workflow. The Read and Write operators are omitted for the sake of brevity.

SENTINEL-1 SNAP OPERATORS

TOPSAR Split → Apply Orbit File → Back-Geocoding → (Enhanced-Spectral-Diversity) → Interferogram → TOPSAR-Deburst → TopoPhaseRemoval → Subset → StampsExport → TerrainCorrection

COSMO-SKYMED SNAP OPERATORS

Subset → Create-Stack → Cross-Correlation → Warp → Interferogram → TopoPhaseRemoval → StampsExport → TerrainCorrection

Regarding, instead, the developed steps about average intensity, local incidence angle and cohere, the following operators are used.

AVERAGE INTENSITY SNAP OPERATORS

Bands Extractor → Band Merge → Band Maths → Subset → Terrain Correction → Write GEOTIFF

LOCAL INCIDENCE ANGLE SNAP OPERATORS

From Interferogram file: Terrain Correction → Write GEOTIFF

COHERENCE SNAP OPERATORS

From Interferogram file: Terrain Correction → Write GEOTIFF

2. AUTOMATED STAMPS PROCESSING

2.1 Preliminary steps

The *PHASE_StaMPS.mlapp* MATLAB application has been developed to improve the *StaMPS* Persistent Scatterer processing. It is aimed to automatically perform both the preparation for the *StaMPS* execution and then the execution itself. The installation of the *StaMPS* packages, as well as its dependencies, is mandatory, and can be done by following the user manual provided by the authors alongside the software (https://github.com/dbekaert/StaMPS). This installation must be done before the execution of the code itself. Remember that this second processing step must be performed in Linux due to limited compatibility of the *StaMPS* software (i.e., it does not work in Windows); so, remember to correctly set the required environment.
The entire procedure has been tested on Ubuntu 20.04, MATLAB 2023b and StaMPS 4.1beta (with TRAIN[3], a toolbox to treat and remove the atmospheric error).

2.2 Required folder structure

The provided MATLAB script has to be left exactly where it is (PHASE_Preprocessing folder), as the last step of the previous code is aimed to correctly move this script in the correct location with its input file and required functions.
Note that I use to have both *StaMPS* and the add-ons installed in the Download folder of the Linux environment, but it's not mandatory. Just check that there are no other files with the same filename in the same installation folder.

2.3 Description of the MATLAB application

This code is aimed at providing an automated procedure that is composed of subsequent steps to perform the *StaMPS* PS analysis and the displacement time series export. It has been tested on MATLAB 2023b, running on a virtual machine, under Linux environment. It is composed of three subsequent steps:

1) Preparation of the required data for the *StaMPS* processing.
2) Definition of all the parameters of the *StaMPS* software.
3) *StaMPS* execution and displacement time series export in an Excel sheet.

Ten panels are present in the application to set all the configurable processing parameters.

a) Preparation: contains the generical paths to the project folder and the one where StaMPS is installed. In addition to that, there are defined the date of the master and the amplitude dispersion threshold. The box is intended to declare if to perform or not this preparation step, as in case of reprocessing.
b) Global Variables: contains all the parameters required for the correct definition of the *StaMPS* input and output files.
c) StaMPS 1 – StaMPS 5: contain all the parameters required for the correct definition of each step of the *StaMPS* processing. The *train box* is intended to declare if to use the TRAIN software or not. Two TRAIN models are currently implemented in the code: linear and GACOS.

d) StaMPS AOI: contains the variables for the definition of the unwrapping spatial reference point, as well as plotting options.
e) Computational Resources: contains the global parameters and path for the correct implementation of the InSAR preprocessing steps.
f) Save/Load: allows the user to save the configured variables and/or load them from a previous saving.
g) Run: contains a button to start the execution of the processing steps.

Note that **all** the variables must be configured as required as input by the *PHASE_StaMPS.mlapp* software. Read the comments for each variable to better understand their meaning.

The usage of this code is based on the previous installation of *StaMPS* and all its additional packages as specified in the *StaMPS* user manual.
It is reminded to check that during the data preparation step **all** the interferograms have a mean amplitude different from zero and no warning arise for each of them. In contrary, it is required to remove them before the *StaMPS* processing to avoid computational errors.

This code contains a *keyboard* command after the automatic creation of the interactive PS plot where the user has to choose the radius and central point for the data export. It is suggested to set a very large radius and select the center of the figure in case all the PS points are relevant for the export. Unfortunately, there is no way to overcome this manual intervention of the user. In addition to that, when also using the TRAIN toolbox, it may be required to press the 'continue' button of the MATLAB editor panel during step 7. Note that this is something *StaMPS*-related and therefore out of the control of this code.

The output of this processing is an Excel (.xlsx and .csv) spreadsheet containing the displacement (in millimeters) for each Persistent Scatterer at each date (of the SAR images). Alongside the displacement output, it is also provided a file containing the atmospheric modelling used in the processing. The output file is created in the way depicted in the below figure.

	[deg]	[deg]	[mm/yr]	Displacements [mm]		
Point	Longitude	Latitude	Mean LOS velocity	Date 1	Date 2	Date 3 …
1						
2						
…						

In case the user wants to compute the "intermediate" graphs, such as the ones for the wrapped and unwrapped phase, it can be done typing the standard *StaMPS* commands directly in the MATLAB command window.

2.4 Possible errors and solutions

During the *StaMPS* processing some problems may arise. This section is intended to provide solutions to the most common ones that have been encountered during the development and test of this software. It is worth mentioning that these errors are not related to the developed application but depends on external sources.

a) *StaMPS* step 5.

During *StaMPS* step 5 the following error might occur.

```
STAMPS: ######################################
STAMPS: ############### Step 5 ###############
STAMPS: ######################################
STAMPS: Directory is PATCH_1

PS_CORRECT_PHASE: Starting
Correcting phase for look angle error...
GETPARM: small_baseline_flag='n'
PS_CORRECT_PHASE: Finished
Error using movefile
Text file busy

Error in stamps (line 462)
        movefile('patch.list','patch.list_old');
```

To solve it, you should change the command *movefile* with *copyfile* inside stamps.mat (lines 462 and 463).

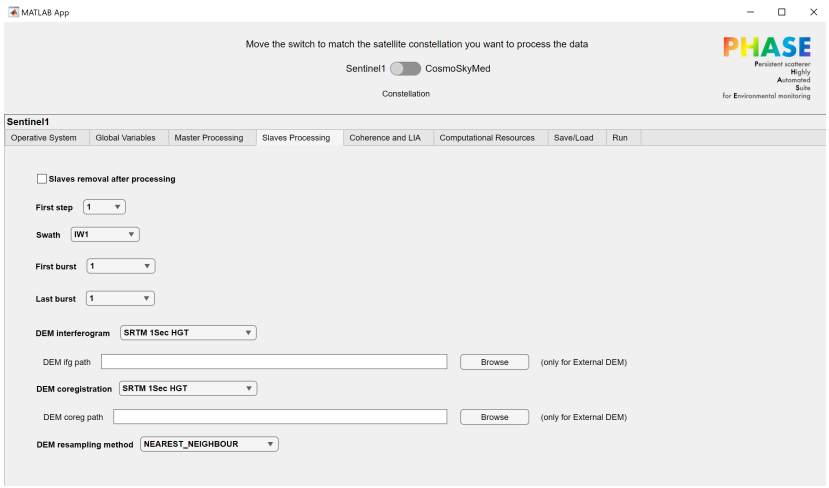
b) GCC and G++

StaMPS requires gcc-7 and g++. Newer versions of Ubuntu (like 22.04) are not natively compatible with *StaMPS* because of this issue. To solve it, you need to install an older version of Ubuntu (like 20.04) or install gcc-7 and g++ on the newer Ubuntu version. For this second option you can refer to:
https://askubuntu.com/questions/1406962/install-gcc7-on-ubuntu-22-04
There might be other packages required to successfully download gcc-7 and g++.

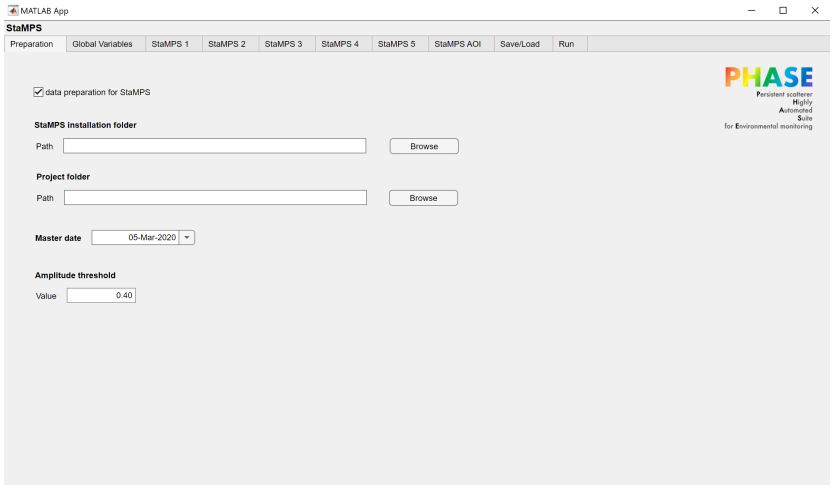
c) PHASE_Preprocessing.mlapp

To process SAR images in a different folder from the ones just done, it is mandatory to firstly close MATLAB, then navigate to the new folder and open the application PHASE_Preprocessing.mlapp to avoid current directory errors.

FIRST PANEL OF THE MATLAB APPLICATIONS



PHASE_Preprocessing.mlapp



PHASE_StaMPS.mlapp

REFERENCES

[1] Foumelis, M., Delgado Blasco, J. M., Desnos, Y. L., Engdahl, M., Fernández, D., Veci, L. Lu, J. and Wong, C. "SNAP -StaMPS Integrated processing for Sentinel-1 Persistent Scatterer Interferometry". In Geoscience and Remote Sensing Symposium (IGARSS), 2018 IEEE International, IEEE.

[2] Hooper, A., A multi-temporal InSAR method incorporating both persistent scatterer and small baseline approaches, Geophys. Res. Lett., 35, L16,302, doi:10.1029/2008GL03465, 2008.

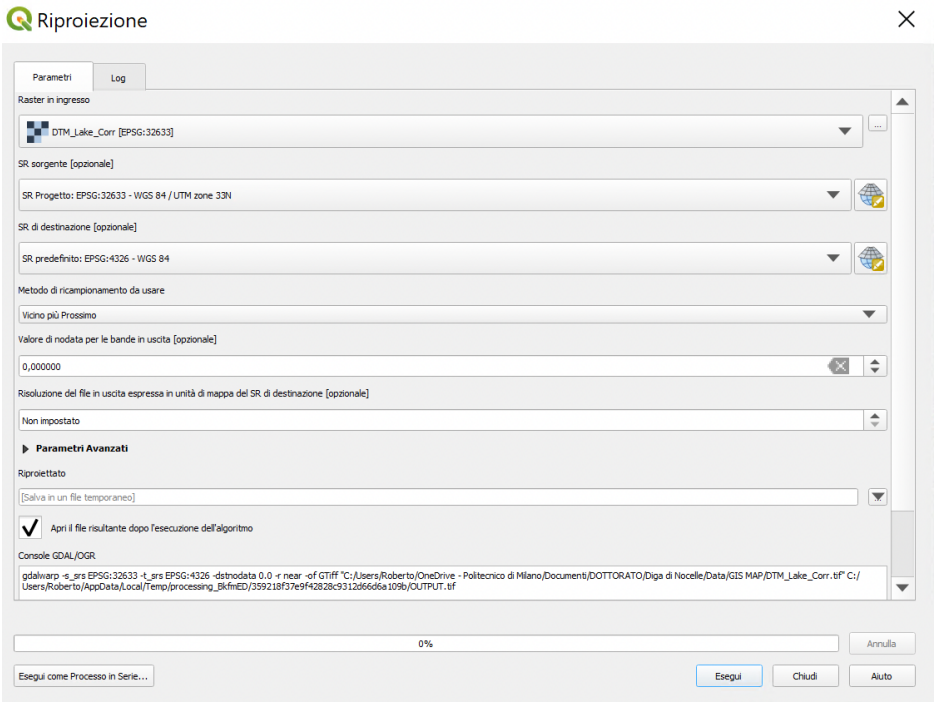
[3] Bekaert, D.P.S., Walters, R.J., Wright, T.J., Hooper, A.J., and Parker, D.J. (2015c), Statistical comparison of InSAR tropospheric correction techniques, Remote Sensing of Environment, doi: 10.1016/j.rse.2015.08.035.

APPENDIX A – PREPROCESSING WITH EXTERNAL DEM

When it is decided to employ an external digital elevation model (DEM) in the *snap2stamps* procedure, it is necessary that it is in the correct format.

FILE TYPE: *.tif*
REFERENCE SYSTEM: WGS 84 – EPSG 4326

To do so, inside QGIS the input DEM is reprojected accordingly to the source's data. In particular, it is important to correctly define the reference system of the desired DEM, as well as defining 0 as *no data value* in the *reproject* toolbox. These concepts are shown in the following image.



Note that for a successful processing it is recommended that the External DEM covers an area that is around 10 % larger than the AOI set for the subset of the SAR images.