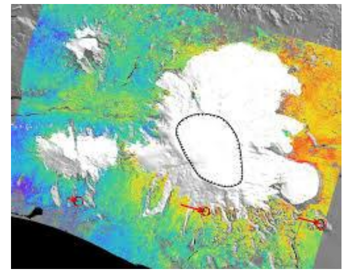

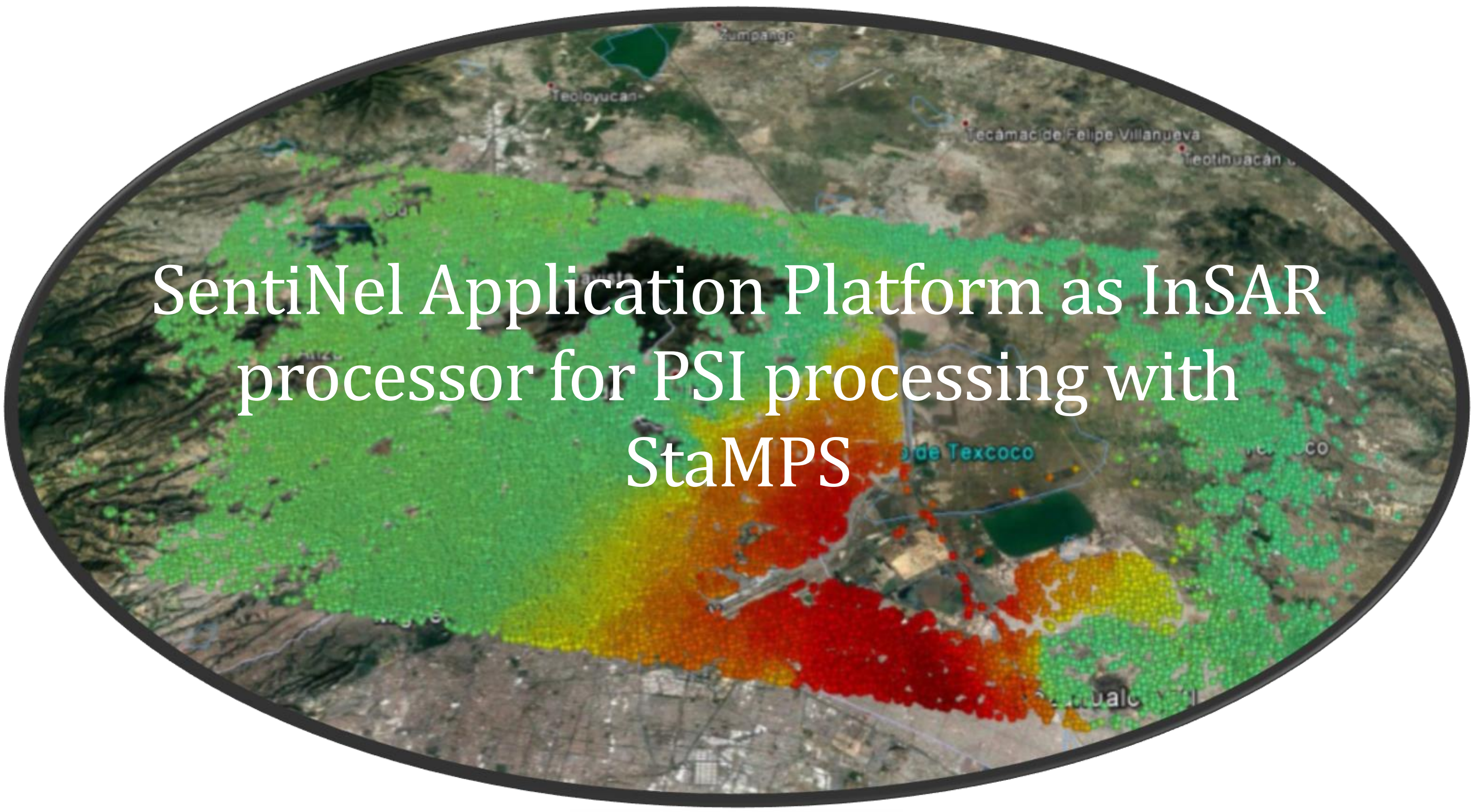

SentiNel Application Platform as InSAR processor for PSI processing with StaMPS

Authors: *José Manuel Delgado Blasco*

Michael Foumelis

Date: *June 2018*

Contents

1 Introduction

The ESA SentiNel Application Platform (SNAP) [1] is the evolution of the Next ESA SAR Toolbox (NEST) developed under an ESA contract which allows users to process satellite data from several Earth Observation satellite missions. In particular, the Sentinel-1 Toolbox is allow the SAR data processing which will be used for our scope.

The Sentinel-1 Toolbox is a multi-platform, multi-sensor, multi-thread software which enables the InSAR, DInSAR and since January 2018 also the interferogram generation compatible with StaMPS for PSI processing [2].

In this manual we want to show step by step the necessary considerations for a successful Sentinel-1 TOPSAR data processing. Demonstrations of the procedure used here has been already shown in [3]. Work was done in collaboration and with coordination with Prof. Andrew Hooper.

For the SAR stripmap interferogram processing we refer a past manuals or trainings done in the past by M. Foumelis and previously by A. Michella and P. Marinkovic using DORIS-NEST. In following versions we will include Stripmap interferogram processing.

When using this software package please refer to :

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

1.1 Installation

As already mentioned, SNAP is a multi-platform software based on java, and the different packages are available for download on the Science Toolbox Exploitation Platform – ESA website http://step.esa.int/main/download/

The version which enables the PSI data processing starts on the 6.0. Previous releases may have issues regarding the production of all the necessary information for the StaMPS PSI processing.

We strongly recommend that which each new release you test the graphs you might have created with previous releases as its total compatibility is not ensured.

After installing SNAP on your computer we suggest you to review the parameters set in the:

- $HOME/snap/bin/gpt.vmoptions and modify the param
  - **–Xmx 12G** (according to your computer set up; i.e –Xmx 512M)
- $HOME/snap/etc/snap.properties
  - #snap.home=
  - #snap.userdir=
  - snap.jai.tileCacheSize = 1024

- snap.jai.defaultTileSize = 512

For the SNAP TOPSAR Interferogram stacking generation you can download the scripts from: https://github.com/mdelgadoblasco/snap2stamps/ which only requires having SNAP and python installed on the system.

Required python modules: pathlib

pip install pathlib

1.2 Hardware requirements

There is not specific hardware requirements, but the interferometric data processing may take longer time to be processed or even fail if the hardware resources of the hosted computer may not be enough. As SNAP is multi-sensor it may work for heritage missions such as ERS or Envisat, where the data volume was limited but it becomes critical when you plan to use Sentinel-1 or another higher resolution SAR data, which data volume is considerable higher than for older SAR missions.

Hence, for Interferogram generation we suggest to have medium-high processing environment. Nowadays there are also processing solutions for research and service developments coming from ESA such as the ESA Research and Service Support which offers the possibility to get a Virtual Machine with user defined resources for temporally processing (see https://wiki.services.eoportal.org/tiki-index.php?page=RSS+CloudToolbox+Service or https://eogrid.esrin.esa.int/cloudtoolbox)

2 Definition of graph workflows using GUI

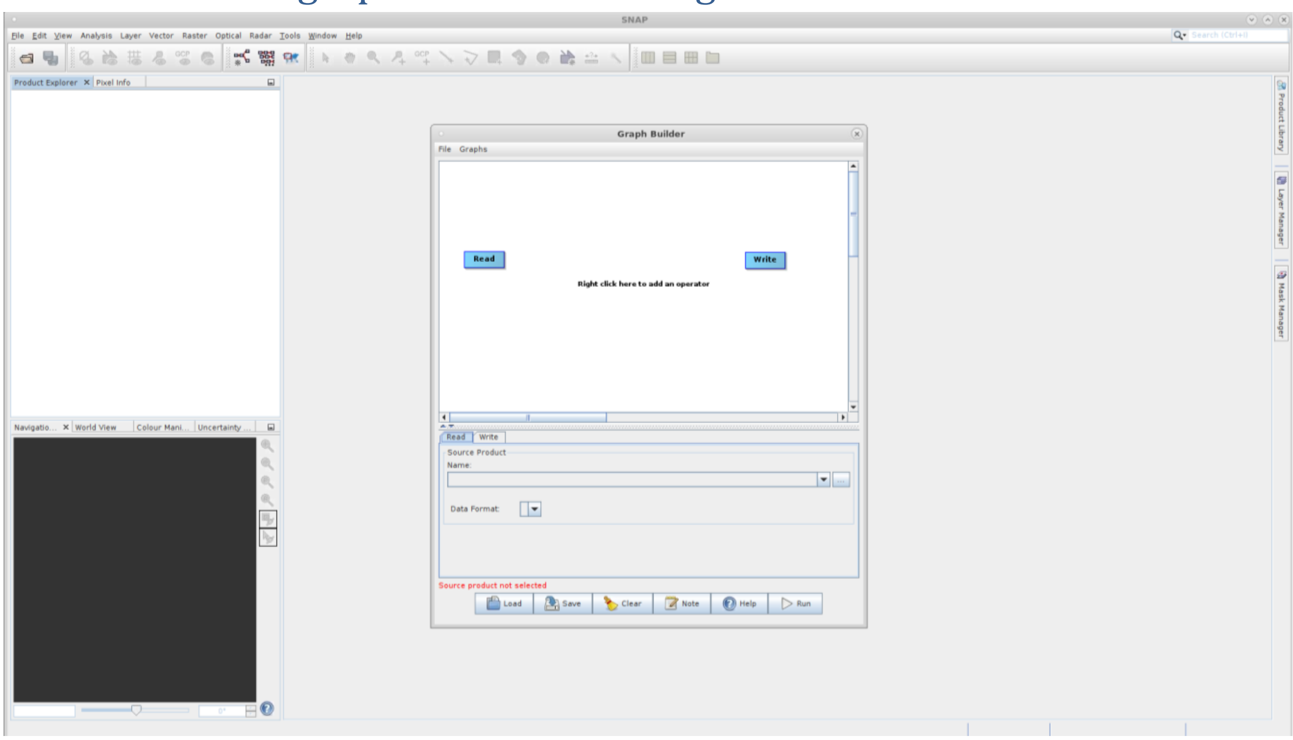

Figure 1. Graph Builder Operator in SNAP GUI

3 Interferogram stacking for StaMPS PSI

3.1 SAR TOPSAR Mode

In this manual we have illustrated how to do interferogram stacking only using one subswath of the Sentinel-1 product. In future versions, we will provide also the scripts for multi-subswath interferometry, which will require more processing time and more performing computing capability.

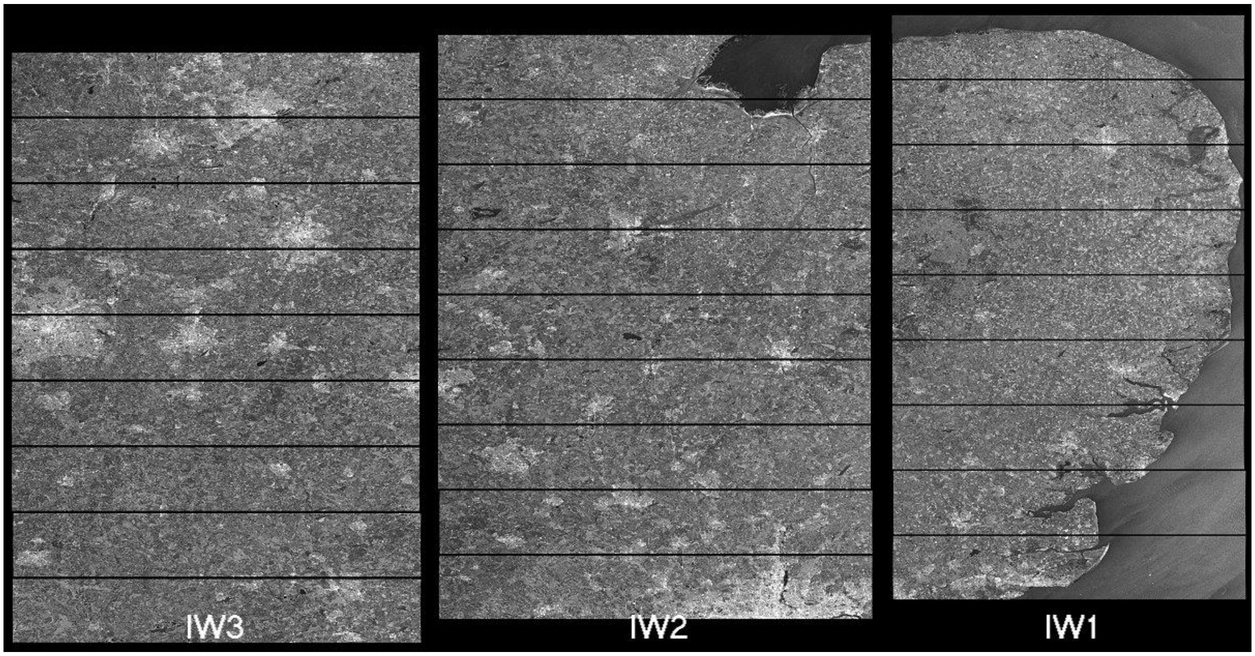

Figure 2. Sentinel-1 IW SLC full image. Visible its 3 subswaths (IW1, IW2 and IW3) with their corresponding burst in which the image can be splitted.

3.1.1 Master Selection and Splitting

The master selection for the Sentinel-1 dataset can be done by using the InSAR Stack Overview available in SNAP.

As the Sentinel-1 TOPSAR SLC image is form by several bursts as illustrated in Fig.2, we strongly recommend to select the Sentinel-1 data among the best masters which covers the desired AOI with a minimum number of burst, if possible within a single subwath (IW) which will optimize the processing time and data volume produced.

Once the master is selected, its subsetting will be done by using the TOPSAR Split operator. The graph shown in Fig. 3 is valid both for master and slave (see Fig.3)

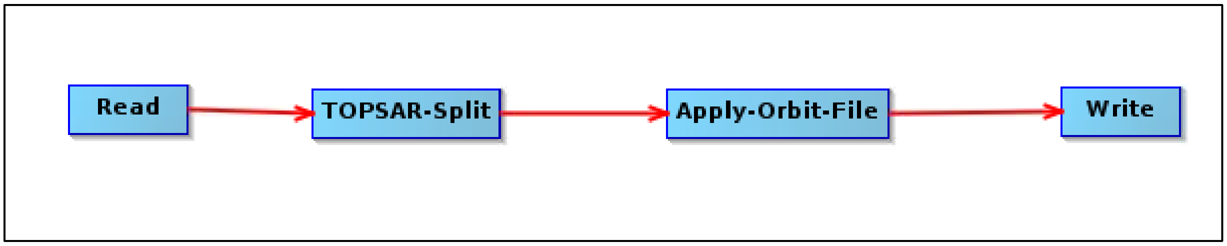

Figure 3. Graph builder with the TOPSAR Split and ApplyOrbits operators.

3.1.2 Slaves split

The slaves subset is done by selecting only the subswath according to the one selected for the master, getting a entire subswath for each slave. In the cases which slaves cover only part of the master AOI, we suggest to download the corresponding slice (if available) which ensure the full master area covered with the assembled slaves. In the same moment we apply the orbit correction by using the Apply Orbits operator (preferably with Precise Orbits and Restituted Orbits only for images acquired in the last 20 days.

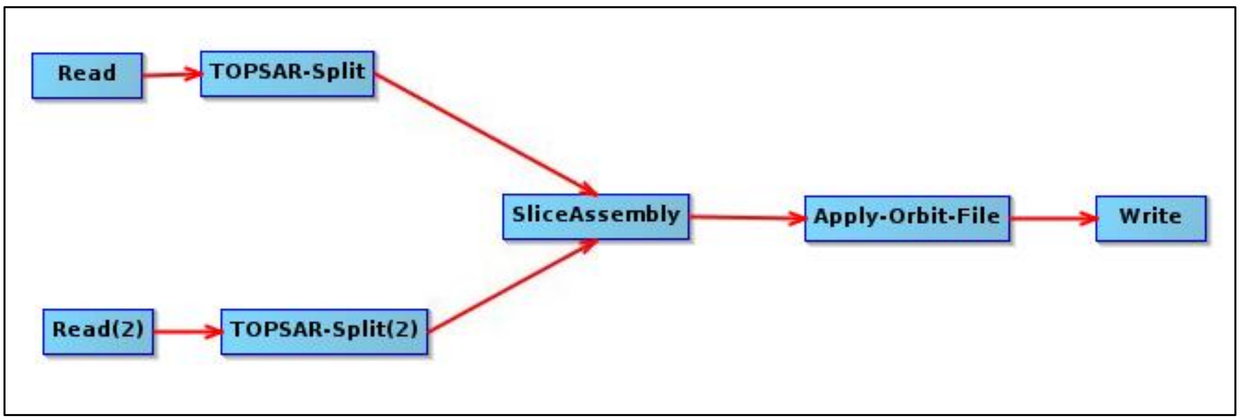

Figure 4. Graph Builder with the definition of a Split-SliceAssemble and ApplyOrbits example, suitable for merging 2 slices of Sentinel-1 image used as slave image.

3.1.3 Coregistration and Interferogram computation

For the coregistration and interferogram computation we proposed a graph which does it all together at a time. This process can take less than 3 minutes of time for processing compressing a single burst and processed on a VM with 8 vCPUs and 32 GB RAM.

In this step, it is only needed a loop for each slave, for a given master image.

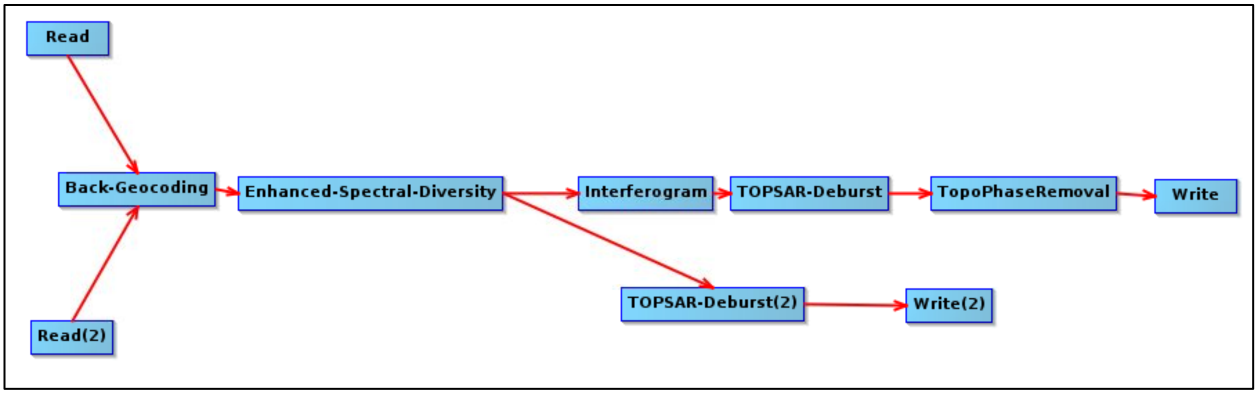

Figure 5. Graph Builder with an example of TOPSAR coregistration and interferogram formation for Sentinel-1 IW SLC splitted products

3.1.4 StaMPS export for PSI

The StaMPS export step is done with the operator with the same name, which inputs are:

i) the coregistered master-slave pair
ii) its corresponding interfeogram with the elevation and orthorectified latitude and longitude bands.

Until this moment this operator only allows you to export in GAMMA format for the PSI processing.

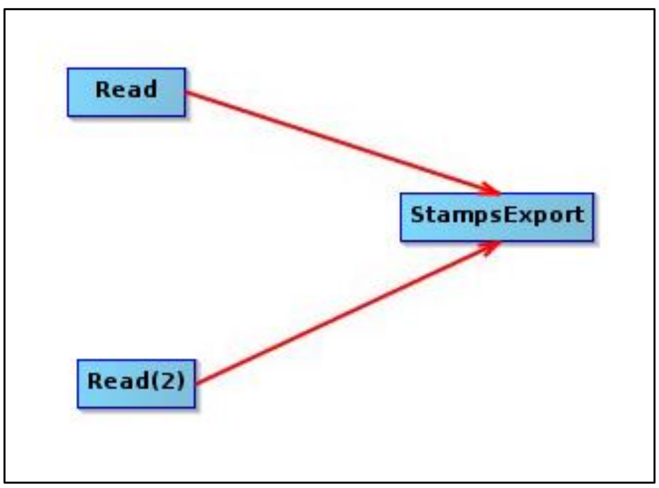

Figure 6. Graph Builder blocks for StaMPS export

4 Running SNAP graphs using the snap2stamps package

In order to produce the stack of interferograms needed for a PSI analysis, we also provide a series of python scripts and processing graphs to be used together with SNAP GPT. These scripts and graphs can be downloaded from the github repository snap2stamps. This package contains the following files:

- **project.conf** – file with parameters and paths needed for the processing
- **slaves_prep.py** – script for slave sortering in the expected folder structure
- **splitting_slaves_logging.py** – script for slave splitting (and assembling is needed) and orbit correction.
- **coreg_ifg_topsar.py** – script for master-slave coregistration and interferometric generation
- **stamps_export.py** – script for ouput data generation in StaMPS compatible format for PSI processing.

The project configuration file should contain the following information :

```
######### CONFIGURATION FILE #####
##################################
# PROJECT DEFINITION
PROJECTFOLDER=/application/wokdir/PROC
GRAPHSFOLDER=/application/workdir/GRAPH
#################################
# PROCESSING PARAMETERS
IW1=IW2
MASTER=
#################################
# AOI BBOX DEFINITION
LONMIN=
LATMIN=
LONMAX=
LATMAX=
#################################
# SNAP GPT
GPTBIN_PATH=/application/pi/snap/bin/gpt
#################################
# COMPUTING RESOURCES TO EMPLOY
CPU=8
CACHE=16
#################################
```

In order to use the aforementioned scripts, it is enough to call python <scriptname> project.conf, as for example:

```
$ python coreg_ifg_topsar.py project.conf
```

Specifically, for each step:

4.1 Preparing slave folders

This step is needed as the following step expects the slaves sorted by acquisition day so it is able to create the slaves folder under the PROJECT path defined on the configuration file project.conf.

```
$ python slaves_prep.py project.conf
```

Requirements: Sentinel-1 data downloaded in zip format on the folder : /<PROJECTFOLDER>/slaves/

4.2 Slave splitting and apply orbit

This step performs the image splitting, taking the VV polarisation channel on the subswath defined on the project.conf configuration file. In case that for a single acquisition day there are 2 slices covering the AOI, they will be assembled using the Slice Assembling operator.

```
$ python splitting_slaves.py project.conf
```

```
Note: current scripts support up to 2 slaves images with same acquisition day (for slice assembling) and
only precise orbits are used. In near future also restituted orbits will be supported.
```

4.3 Coregistration and Interferogram generation

This step is the most time consuming and it performs the coregistration and interferogram computation (currently supports only single subswath processing) for each master-slave pair. Results are saved on coreg and ifg folders under the PROJECT folder.

```
$ python coreg_ifg_topsar.py project.conf
```

```
Note: SRTM1 arc second is used for both Backgeocoding and TopoPhaseRemoval computation. In the future more
DEM will be supported via configuration file.
```

4.4 StaMPS export

In order to run this step the command should be

```
$ python stamps_export.py project.conf
```

The data final output structure after this step should contain these four folders: rslc, diff0, geo and dem, as indicated below:

```
pi@CToolbox_IPT_Chris_Stewart:/application/.proctmp/Cairo_asc/PSIexport/PSI$ ls -l
drwxrwxr-x  7 pi pi  4096 Jun 22 10:03 ./
drwxrwxr-x  3 pi pi  4096 Jun 21 15:25 ../
drwxrwxr-x  2 pi pi  4096 Jun 21 15:25 dem/
drwxrwxr-x  2 pi pi 20480 Jun 21 17:31 diff0/
drwxrwxr-x  2 pi pi  4096 Jun 21 15:25 geo/
drwxrwxr-x  2 pi pi 12288 Jun 21 17:31 rslc/
```

4.5 Matlab preparation for StaMPS PSI : mt_prep_snap

As last step before running StaMPS, data must be prepared and PS candidates should be identified. Here we use the provided by Andy Hooper's script customised for SNAP interferograms generated.

```
$ mt_prep_snap 20150419 /application/workdir/Rome/export/PSI/INSAR_20150419 0.35 3 3
```

After running the mt_prep_snap script, data is ready to run StaMPS on it. The final folder structure and content should looks like here below:

```
pi@CToolbox:/application/workdir/Rome/export/PSI/INSAR_20150419$ ls -l
drwxrwxr-x 22 pi pi  4096 Jun 21 20:41 ./
drwxrwxr-x  3 pi pi  4096 Jun 21 20:22 ../
-rw-rw-r--  1 pi pi  6880 Jun 21 20:22 calamp.in
-rw-rw-r--  1 pi pi  7558 Jun 21 20:41 calamp.out
drwxrwxr-x  2 pi pi  4096 Jun 21 15:25 dem/
drwxrwxr-x  2 pi pi 20480 Jun 21 17:31 diff0/
drwxrwxr-x  2 pi pi  4096 Jun 21 15:25 geo/
-rw-rw-r--  1 pi pi     5 Jun 21 20:22 len.txt
drwxrwxr-x  2 pi pi  4096 Jun 21 20:46 PATCH_1/
drwxrwxr-x  2 pi pi  4096 Jun 21 20:59 PATCH_2/
drwxrwxr-x  2 pi pi  4096 Jun 21 21:22 PATCH_3/
drwxrwxr-x  2 pi pi  4096 Jun 21 21:41 PATCH_4/
drwxrwxr-x  2 pi pi  4096 Jun 21 22:01 PATCH_5/
```

```
drwxrwxr-x  2 pi pi  4096 Jun 21 22:19 PATCH_6/
drwxrwxr-x  2 pi pi  4096 Jun 21 22:40 PATCH_7/
drwxrwxr-x  2 pi pi  4096 Jun 21 23:00 PATCH_8/
drwxrwxr-x  2 pi pi  4096 Jun 21 23:20 PATCH_9/
-rw-rw-r--  1 pi pi   135 Jun 21 20:41 patch.list
-rw-rw-r--  1 pi pi     6 Jun 21 20:22 processor.txt
-rw-rw-r--  1 pi pi    89 Jun 21 20:41 pscdem.in
-rw-rw-r--  1 pi pi   162 Jun 21 20:41 psclonlat.in
-rw-rw-r--  1 pi pi  7656 Jun 21 20:41 pscphase.in
-rw-rw-r--  1 pi pi    84 Jun 21 20:22 rsc.txt
drwxrwxr-x  2 pi pi 12288 Jun 21 17:31 rslc/
-rw-rw-r--  1 pi pi  7568 Jun 21 20:41 selpsc.in
-rw-rw-r--  1 pi pi     6 Jun 21 20:22 width.txt
```

5 Future modifications

There is already a list of identified future updates.

Community contributions are welcomed.

1) Add more sensor support. Specifically on STRIMAP SAR datasets

2) External DEM support configurable via python scripts. Already possible modifying provided graphs directly.

3) Applied Orbit configurable via python scripts. Already possible modifying provided graphs directly.

4) Include script for downloading orbit files directly from ESA server to avoid problem with 3rd party dependencies.

5) Include script for sentinel-1 data download from the Copernicus Open Data Hub (registration needed by the user)

6) Multi subswath integration. Current scripts do not support 2 or 3 subswath interferograms.

7) Prepare scripts for StaMPS SBAS

6 History

1.0 – Scripts provided for single subwath interferogram stacking. Single slave image supported. Precise Orbits and STRM DEM 1 Arc second used. Currently the user should define a bounding box for AOI definition.

1.1 – Support slice assembling with a maximum of 2 slaves per acquisition day.

1.2 – Full burst processing allowed. No bounding box definition needed. New option of no subseting after coregistration or interferogram to be performed.

7 References:

[1] Desnos, Y. L., Foumelis, M., Engdahl, M., Mathieu, P. P., Palazzo, F., Ramoino, F., & Zmuda, A. (2016, July). “Scientific Exploitation of Sentinel-1 within ESA's SEOM programme element”. In *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International* (pp. 3878-3881). 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] Foumelis, Michael, 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