Towards SNAP-StaMPS
Automatic PSI Processing Service for Research Applications on ESA GEP Cloud Infrastructure
Motivation

• Since the launch of Sentinel-1A, few open source InSAR processors were available for generating interferogram stacks compatible with StaMPS*.

• ESA Sentinel-1 toolbox had almost all the functionalities to fill that gap, but compatibility with StaMPS it was not initially designed.

• On the ESA STEP forum (https://forum.step.esa.int/) many threads, by the user community, address issues while trying to use SNAP as a pre-processing tool for PSI analysis.

→ This work was born as answer to this user community need

   Design an optimal DInSAR processing chain & ensure compatibility with StaMPS PSI

---

SNAP

- The common architecture for all Sentinel Toolboxes and SMOS Toolbox is called Sentinel Application Platform (SNAP).

- SNAP architecture is ideal for Earth Observation processing and analysis due the following technological innovations: Extensibility, Portability, Modular Rich Client Platform, Generic EO Data Abstraction, Tiled Memory Management and a Graph Processing Framework.

Activity initially funded through SEOM element of ESA’s EOEP-4 (www.seom.esa.int)
 Sentinel Application Platform | SNAP

- **SAR Toolbox (S1TBX)**
  - Scientific toolbox for the handling and post-processing of data products from Sentinel-1 SAR mission

- **High Resolution Optical Toolbox (S2TBX)**
  - Toolbox for the visualisation, analysis and post-processing of data products from Sentinel-2 multi-spectral optical data

- **Medium Resolution Optical Toolbox (S3TBX)**
  - Toolbox for the processing and analysis of Sentinel 3 OLCI and SLSTR

- **Developer forum**
  - Requirements addressing a common platform issues
  - Define the platform roadmap
  - Coordinate horizontal activities across the three toolboxes
Multi-Mission Scientific Platform
Development Consortia
SNAP Development History

Built on prior toolbox development

nest
next esa sar toolbox

BEAM
Empower the EO community to better exploit the large archives of the Sentinels and heritage missions in both research and operational usage.

Evolve the architecture to ensure that the software will be capable of supporting the large data products and ever growing volumes of EO data.
− Developed as open source software
− Common Java core framework
− Joint development plan for Sentinel toolboxes
− Interchangeable Java/Python plugins
− Portable engine to Cloud infrastructure
− Single installer
TOPSAR InSAR Chain

Support from SEOM R&D projects

**S1-INSARAP: SENTINEL-1 INSAR PERFORMANCE STUDY WITH TOPS**

An ESA project kicked off in March 2014 after successful contract negotiations (www.seom.esa.int).

"Validation and scientific exploitation of the interferometric performance of TOPS mode on Sentinel-1 mission"

- Full exploitation of S-1 mission interferometric capacity
- Development of advanced algorithms for TOPS data
- Demonstrate continuity of ESA’s C-band SAR observations
SNAP Sentinel-1 Interferometry | New Zealand Earthquake

Terrain Corrected
Wrapped Interferogram

ASCENDING
S1A_20161103
S1A_20161115

DESCENDING
S1A_20161010
S1A_20161116
SNAP Download page

Access to Beta versions for testing

Technical documentation for both end-users and developers

Step-by-step tutorials including

YouTube videos

Technical forum, gathering user feedback and communicating results
SNAP
All-in-One Environment

This tool window is used to manipulate the colouring of images shown in an image view. Right now, there is no selected image view.
ESA Grid Processing on Demand | G-POD

- ERS 1/2 SAR Data Archive
- ENVISAT ASAR Data Archive
- SENTINEL-1 TOPS Data

Direct access to **G-POD** and **Virtual Archive 4**
ERS 1/2 and ENVISAT SAR data

**Sentinel-1 Data Hub** connected to G-POD
- Supported catalogue search from G-POD GUI
- Automatic data retrieval from G-POD Worker Nodes
- Data cache of latest downloaded S-1 products
ESA G-POD service is provided by RSS aiming to support the Earth Observation community in data exploitation.
TerraDue Cloud ToolBox

The service offers configurable Virtual Machines (VM) tailored to scientific users’ requirements.


Access the data packages retrieved from the GEP Portal
SNAP DInSAR Processing Chain

- Split IW Subswath (incl. Polarization) over AOI
- Update Orbit State Vectors
- Back-geocoding (DEM-assisted coregistration)
- Correct for Range and Azimuth Shifts (ESD algorithm)
- Interferogram Generation (incl. Coherence)
- Goldstein Phase Filtering
- Phase Unwrapping (SNAPHU)
- Convert Phase to Displacement
- Terrain Correction Geocoding

TOPS Coregistration
SNAP DInSAR Pre-Processing Steps for StaMPS

- Split IW Subswath (incl. Polarization) over AOI
- Update Orbit State Vectors
- Back-geocoding (DEM-assisted coregistration)
- Correct for Range and Azimuth Shifts (ESD algorithm)
- Interferogram Generation (incl. Coherence)
- Goldstein Phase Filtering
- Phase Unwrapping (SNAPHU)
- Convert Phase to Displacement
- Terrain Correction - Geocoding

TOPS Coregistration
SNAP DInSAR Demo | Input Dataset

- A set of **Sentinel-1A SLCs** $YYYMMDDTHHMMSS$
  
  S1A_IW_SLC__1SDV_20190817T171546_20190817T171613_028612_033C97_F4AE
  S1A_IW_SLC__1SDV_20190829T171546_20190829T171613_028787_0342BB_E70D
  
  [*downloadable via Copernicus OpenHub, PEPS, ASF, etc.*]

- Sentinel-1 **Precise Orbits** (PODs) for the corresponding S1A dates (*.*.EOF files are automatically downloaded via https://qc.sentinel1.eo.esa.int)

- **Digital Elevation Model** (DEM) dataset from SRTM 3 arc-sec covering the Area of Interest (automatically downloaded from the ESA SNAP repository)

**VM configuration on ESA G-POD**

OS: Ubuntu Linux

Number of Cores: 8

RAM: 32 GB

Dedicated Storage: 3 TB
SNAP DInSAR Processing Scheme

Read S1 SLC products
SNAP DInSAR Processing Scheme

Master SLC Splitting (burst-level)
SNAP DInSAR Processing Scheme

Master SLC Splitting (burst-level)
SNAP DInSAR Processing Scheme

Master SLC Splitting (burst-level)
SNAP DInSAR Processing Scheme

Update Orbits (Restituted or Precise)
SNAP DInSAR Processing Scheme

Update Orbits (Restituted or Precise)
SNAP DInSAR Processing Scheme

Update Orbits (Restituted or Precise)
SNAP DInSAR Processing Scheme

Splitting & Orbits for Slave SLCs
SNAP DInSAR Processing Scheme

Splitting & Orbits for Slave SLCs
SNAP DInSAR Processing Scheme

Splitting & Orbits for Slave SLCs
SNAP DInSAR Processing Scheme

Splitting & Orbits for Slave SLCs
SNAP DInSAR Processing Scheme

Splitting & Orbits for Slave SLCs
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Co-registration & Interferogram generation
SNAP DInSAR Processing Scheme

Overview of processing outputs
SNAP InSAR processing
Demo
ESA Sentinel Application Platform | SNAP

SNAP Interferometric Processing | Manual vs Batch mode

snap2stamps | Python Scripts

StaMPS PSI Processing Scheme

SNAP-StaMPS PSI service on GEP
Several parameters (in configuration file) need to be set for running the scripts successfully:

- Project folder
- CPU and Cache memory specification
- Path to the SNAP Graph Processing Tool (GPT)
- Subswath(s) to process

snap2stamps | Software Package

- Created as a response to the user community needs (SNAP Forum)
- Open source and available on Zenodo repository [DOI 10.5281/zenodo.1308835]
- Python scripts and pre-define xml graphs which uses SNAP to create stacks of interferograms compatible with StaMPS PSI
- Currently supports Sentinel-1 TOPSAR SLC data
Automated SNAP Sentinel-1 DInSAR processing for StaMPS PSI with open source tools

Jose Manuel Delgado Blasco, Michael Fournier

This software package provides a set of python scripts that call routines from the ESA Sentinel Application Platform (SNAP) to allow for automatic interferogram stacking that is compatible with StaMPS PSI.

The initial version works with Sentinel-1 IW SLC products.


Processing Steps:

1. Sort slave images in separated folders corresponding to their acquisition time.
2. Slave splitting (incl. slice assembling, if needed) and updating orbits (Precise or Restituted).
3. Coregistration and interferogram generation per slave and subswath. A subset option using a bounding box is now available.
4. StaMPS export is done providing for each pair coregistered stack and interferogram generated StaMPS compatible products.
A minimum of two bursts are required for snap2stamps processing.

For 2-burst image approx. 2-3 min of processing time area needed per interferogram. Exponential increase by adding more bursts.
Auxiliary files as DEM and Orbit State Vectors are automatically downloaded by SNAP. Subsetting over an AOI using a bounding box is supported in version 1.0.1.

* We suggest to proceed with master scene selection and sub-setting using SNAP GUI.

**Image Splitting and Update of Orbits**

- Master scene
- Slave scenes
TOPS Co-registration and Interferogram formation

Both amplitude coregistered master-slaves stack and interferograms, including heights and orthorectified latitude and longitude images are saved as output.

SNAP export to StaMPS format
• `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 output data generation in StaMPS compatible format for PSI processing.
##### CONFIGURATION FILE ####
#############################
# PROJECT DEFINITION
PROJECT_FOLDER=/application/workdir/PROC_dir
GRAPHSFOLDER=/application/graphs
#############################
# PROCESSING PARAMTERS
IW1=IW2
MASTER=/application/workdir/PROC_dir/master/S1A_IW_SLC__1SDV_20150402T155633_20150402T155700_022180_02662B_7085_split_Orb.dim
#################################
# AOI BBOX DEFINITION
LONMIN=
LATMIN=
LONMAX=
LATMAX=
#################################
# SNAP GPT
GPTBIN_PATH=/application/pi/snap/bin/gpt
#################################
# COMPUTING RESOURCES TO EMPLOY
CPU=8
CACHE=16G
#################################
Preparing slave folders

$ python slaves_prep.py project.conf

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

Slave splitting and apply orbit

$ 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.

Coregistration and Interferogram generation

$ 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.

StaMPS export

$ python stamps_export.py project.conf
snap2stamps | Outputs

Parent directory

```
drwxrwxr-x 11 pl pt 4096 0kt 9 08:50 ./
drwxrwxr-x  8 pl pt 4096 0kt 8 23:53 ../
drwxrwxr-x  5 pl pt 4096 0kt 9 01:09 coreg/
drwxrwxr-x  2 pl pt 4096 0kt 8 23:58 graphs/
drwxrwxr-x  2 pl pt 4096 0kt 8 08:52 INSAR_20190817/
drwxrwxr-x  2 pl pt 4096 0kt 8 23:35 master/
drwxrwxr-x  3 pl pt 4096 0kt 8 23:53 MasterSplit/
drwxrwxr-x  6 pl pt 4096 0kt 9 00:29 slaves/
drwxrwxr-x  5 pl pt 4096 0kt 8 23:53 split/
```

StaMPS processing directory

```
(base) pi@Toolbox-Foumelis:/application2/workdir/mdis/zip$ ll INSAR_20190817/
total 24
drwxrwxr-x  6 pl pt 4096 0kt 9 08:52 ./
drwxrwxr-x 11 pl pt 4096 0kt 9 08:50 ../
drwxrwxr-x  2 pl pt 4096 0kt 9 00:29 coreg/
drwxrwxr-x  2 pl pt 4096 0kt 8 23:35 master/
drwxrwxr-x  2 pl pt 4096 0kt 9 08:54 dem/
drwxrwxr-x  2 pl pt 4096 0kt 9 08:56 diff0/
drwxrwxr-x  2 pl pt 4096 0kt 9 08:52 geo/
drwxrwxr-x  2 pl pt 4096 0kt 9 08:56 rslc/
```

- `./rslc/yyyyymmdd.rslc`
  Master SLC and a resampled SLC for every slave image

- `./rslc/yyyyymmdd.slc.par`
  SLC parameter file for the master scene

- `./diff0/yyyyymmdd.diff`
  A single master interferogram for every slave image

- `./geo/yyyyymmdd lon` & `./geo/yyyyymmdd.lat`
  Longitude and latitude images for every pixel in the master RDC geometry

- `./geo/yyyyymmdd dem.rdc`
  DEM in master RDC coordinates

- `./diff0/yyyyymmdd.base`
  A baseline file for every interferogram pair
In the framework of snap2stamps development several bugs on SNAP v5 were identified and corrected (other issues to be addressed in future versions).

SNAP from v6 onwards is able to provide interferogram stacks compatible with StaMPS PSI.

End-to-end PSI processing using SNAP & StaMPS PSI showed consistent results with already published studies.

The open and free snap2stamps software package provides the community with an easy way to automatize the single master bulk DInSAR processing.

Further developments shall be done to fulfill other user community needs.
- Add more sensor support, specifically for STRIMAP SAR data
- External DEM support configurable via python scripts. Already possible by directly modifying provided graphs.
- Applied Orbit configurable via python scripts. Already possible by directly modifying provided graphs.
- Include script for downloading orbit files directly from ESA server to avoid problem with 3rd party dependencies.
- Include script for Sentinel-1 data automatic download via the Copernicus Open Data Hub and PEPS (registration needed by the user).
- Multi sub-swath integration (sub-swath merging). Current scripts support single sub-swath interferometric processing.
- Prepare scripts for StaMPS SBAS (dependency on SNAP development)
snap2stamps python scripts
Demo
ESA Sentinel Application Platform | SNAP
SNAP Interferometric Processing | Manual vs Batch mode
snap2stamps | Python Scripts
StaMPS PSI Processing Scheme
SNAP-StaMPS PSI service on GEP
StaMPS/MTI Software

Since July 2018 (version announced at IGARSS 2018) the software is distributed via GitHub repository.

A software package to extract ground displacements from time series of synthetic aperture radar (SAR) acquisitions.

The original version was developed at Stanford University but subsequent development has taken place at the University of Iceland, Delft University of Technology and the University of Leeds.

The package incorporates persistent scatterer and small baseline methods plus an option to combine both approaches.

**Andy Hooper**
School of Earth and Environment
University of Leeds
Leeds LS2 9JT

15th August, 2018
A new beta release version of StaMPS/MTI (version 4.1b1) is available from Github (extract with `tar -zxvf`).

The manual of StaMPS/MTI (version 4.1b1) is available as a [pdf file](#).

12th September, 2013
A new beta release version of StaMPS/MTI (version 3.3b1) is available as [StaMPS_v3.3b1.tar.gz](#) (extract with `tar -zxvf`).

The manual of StaMPS/MTI (version 3.3b1) is available as a [pdf file](#).

24th November, 2010
A new release version of StaMPS/MTI (version 3.2) is available as a [tar.gz file](#) (extract with `tar -zxvf`).

* StaMPS/MTI version 3.2 is updated to version 3.2.1 on 26th November, 2010.

3rd March, 2010
A new beta version of StaMPS/MTI (version 3.2b4) is available as a [tar.gz file](#) (extract with `tar -zxvf`).

9th December, 2009
A new beta version of StaMPS/MTI (version 3.2b3) is available as a [tar.gz file](#) (extract with `tar -zxvf`). An updated manual is included in the zip file.

15th July, 2009
StaMPS/MTI (version 3.1) is available as a [tar.gz file](#) (uncompress with `gunzip`, then extract with `tar -zxvf`). This software may be downloaded freely for non-commercial applications.

If you use this code, please join the user group [MAINSAR](#) to be informed of any updates/Issues. Also, please post any questions or advice to this group rather than directly to me.

**Andy Hooper**
In the **INSAR_masterdate** directory run **mt_prep_snap** command

For example:

```bash
mt_prep_snap 0.4 3 3 50 200
```

where
- **0.4** = amplitude dispersion (0.4-0.42 are reasonable values)
- **3** = number of patches in range (default 1)
- **3** = number of patches in azimuth, (default 1)
- **50** = overlapping pixels between patches in range (default 50)
- **200** = overlapping pixels between patches in azimuth (default 200)
Ingestion of SNAP outputs into StaMPS

Preparation for StaMPS PSI inputs: `mt_prep_snap` command

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

```
pi@CToolbox:/application/workdir/Rome/export/PSI/INSAR_20150419$ ls -l
```

```bash
pi@CToolbox:/application/workdir/Rome/export/PSI/INSAR_20150419$ ls -l
```
StaMPS Processing Parameters

The parameters that control the processing can be viewed in matlab

>> getparm

Modify any parameters from the default

>> setparm('param_name', param_value)

Setting param_value to nan resets the parameter to the default value.

Commands for running StaMPS processing with or without APS correction (step 8)

>> stamps(1,7)

or with APS

>> stamps(1,8)
The Toolbox for Reducing Atmospheric InSAR Noise – TRAIN – is developed in an effort to include current state of the art tropospheric correction methods into the default InSAR processing chain. Initial development was performed at the University of Leeds. TRAIN toolbox is integrated in the default processing chain of StaMPS.

Spectrometer - MERIS (ENVISAT data) & MODIS Weather model - ERA-I, MERRA, MERRA-2, GACOS Weather Research and Forecasting Model (WRF) Power-law correction for tropospheric delays Linear phase-topography correction

SNAP-StaMPS vs ESA InSARap project | Mexico City

PSI Sentinel-1 results over pilot site of Mexico City obtained with the DLR-HR TAXI PSI processor

SNAP-StaMPS PSI processing

ESA InSARap project
SNAP-StaMPS integrated Sentinel-1 PSI | Rome
StaMPS PSI Processing Scheme
Demo
ESA Sentinel Application Platform | SNAP

SNAP Interferometric Processing | Manual vs Batch mode

snap2stamps | Python Scripts

StaMPS PSI Processing Scheme

SNAP-StaMPS PSI service on GEP
SNAP-StaMPS PSI Service
Timeline of GEP Integration

- **June 2018**: SNAP pre-processing code shared on GitHub (incl. output PSI dataset)
- **Jan 2019**: Code preparation
- **May 2019**: Integration on GEP
- **Sept 2019**: Service pre-operations
- **Oct 2019**: Open consolidated service
- **Nov 2019**: Outputs verification & performance assessment
- **Today**: Systematic service

Want to apply for the GEP Early Adopters Programme?
contact@geohazards-tep.eu
It’s a two step process.

The first consists in setting-up a data processing pipeline to generate the interferogram stack:

- You select a stack of Sentinel-1 SLC with the same orbit
- You select the swath(s)
- You select a tag for the stack
- The data pipeline generates a set of data items

In the second step the interferogram stack is channeled to the PSI pipeline for SNAP-StaMPS PSI time series analysis.
The goal is to create data processing pipelines to:

- Generate stacks of interferograms derived from Sentinel-1 SLC data
- Generate PSI time series with StaMPS out of those interferograms stacks

**Interferograms stack data processing pipeline**

For each slave paired with a defined master, the data processing pipeline generates one processing request per swath.

**PSI processing pipeline**

For each swath stack of interferograms, the data processing pipeline triggers a StaMPS PSI processing request.
SNAP-StaMPS PSI | Processing Modes on GEP

Systematic processing
- Systematically generate stacks of interferograms derived from Sentinel-1 SLC data
- Automatically generate PS time-series with StaMPS out of those interferograms stacks when certain conditions are reached (e.g. size of the stack)

On-demand processing
- Via Geobrowser or API, create stack of interferograms with a defined list of Sentinel-1 SLC products
- Wait for it’s automatic production
- Via Geobrowser or API, select a stack of interferograms and generate PS time-series with StaMPS

Advanced usage
- Via Jupyter Notebooks or scripting, select a stack of interferograms, do its stage-in and do the StaMPS processing steps interactively allowing the incremental processing and check on intermediate data
The service outputs correspond to the generic outputs of StaMPS, which is a set of files in ASCII format, for the estimated velocities, corresponding uncertainties and the actual displacement time series.

### Outputs

- **ps_mean_v.xy** (longitude, latitude, mean velocity deformation information)
- **ps_data.xy** (longitude, latitude, mean_v, mean_v_std, dem height, dem_error, inc_angle)
- **ps_u-dm.1.xy** : longitude, latitude, phase minus dem error and master aps in mm
- **ps_u-dm.N.xy**, with N equal of each interferogram

Default corrections applied include DEM error and master APS (v-dm), while atmospheric screen correction is optional.

**Outputs at reduced spatial resolution of 100m**
SNAP-StaMPS PSI | GEP User Interface (on-going development)
SNAP-StaMPS PSI | GEP User Interface (on-going development)
SNAP-StaMPS service on GEP
Demo
Thank you