

PySAR v1.0 (A Python Module for InSAR Time-Series Analysis)

Author: Heresh Fattahi, University of Miami

Automatic processing:

To automatic generation of the time-series using PySAR a template file(ProjectName.template) with the following options is required:

When the template file is ready then run this command:

```
PySAR.py ProjectName.template
```

```
#####
```

```
PySAR.inputdata = path to the unwrapped interferograms
```

```
PySAR.CorFiles = path to the coherence files
```

```
PySAR.dem = path to the dem in the coordinate system of interferograms (radar or geo)
```

```
PySAR.seed.yx= y, x [To reference all interferograms to the same pixel]
```

```
PySAR.topo_error = yes or no [Default: no]
```

```
PySAR.orbit_error = yes or no [Default: no]
```

```
PySAR.orbit_error.method = To correct the orbital error any of the following options can be used
```

```
'quadratic', 'plane', 'quadratic_range', 'quadratic_azimuth', 'plane_range', 'plane_azimuth', 'baselineCor', 'BaseTropCor'
```

```
PySAR.tropospheric_delay= yes or no [Default: no]
```

```
PySAR.tropospheric_delay.method=height-correlation or pyaps
```

```
PySAR.trop.polyOrder = 1,2,3 [Default: 1]
```

```
PySAR.mask=yes or no [Default: yes] uses the temporal coherence for masking velocity
```

PySAR.mask.threshold = [Default: 0.7]

#####

Following is a sample template file:

#####

```
PySAR.inputdata=/mgg/PROCESS/SanAndreasT356EnvD/DONE/filt*unw
PySAR.CorFiles = /mgg/PROCESS/SanAndreasT356EnvD/ DONE/filt*0*.cor
PySAR.dem = /mgg/TSSAR/TestPySAR/radar_8rlks.hgt
PySAR.seed.yx=257 , 151
PySAR.topo_error = yes
PySAR.orbit_error = yes
PySAR.orbit_error.method = baselineCor
PySAR.tropospheric_delay= yes
PySAR.tropospheric_delay.method=height-correlation
PySAR.trop.polyOrder = 1
PySAR.mask=yes
PySAR.mask.threshold = 0.8
```

#####

Interactive processing:

Any of the following functions can be used for interactive processing. Running each function generates simple explanation and examples.

+++++

Baseline_error_estimation.py :

Estimating the parameters of baseline error including horizontal and vertical baseline error and their rates error. (Slightly modified implementation of Gourmelen et. al., 2010)

+++++

Baseline_trop_correction.py:

Joint estimation of the baseline error and the tropospheric delay correlated with topography. (An implementation of Min-Jeong Jo et. al, 2010)

+++++

Correlation_with_DEM.py

Calculates the correlation of DEM with the velocity field. (No correction is applied)

+++++

InSARvsGPS_velocity.py

Compares InSAR velocity with GPS velocity in LOS.

+++++

LOD_Correction.py

+++++

Masking.py

To mask out some area from the velocity field. Usually to mask out the incoherent pixels based on temporal coherence.

+++++

Matching.py

To match two adjacent velocity fields with common area. Both should be geocoded.

+++++

RemovePlane.py

To remove a surface (quadratic or plane) in range, azimuth or both directions from time-series epochs or interferograms.

+++++

SeedData.py

To reference all interferograms to the same pixel.

+++++

convert2mat.py

To convert the velocity field to a matlab structure file.

+++++

diff.py

To calculate the difference of two PySAR products.

+++++

generateMask.py

To generate a mask file.

+++++

geocode.py

To geocode the PySAR products (time-series, velocity, temporal coherence, ...) using roi_pac scripts. geomap.trans file is required.

+++++

igram_inversion.py

To invert the interferograms using L1 and L2 norm minimization. (L1 is still under test)

+++++

info.py

gives general information about the PySAR hdf5 files.

+++++

load_data.py

To load the interferograms and coherence maps in an HDF5 file. It also generates Mask.h5 which is 1 for those pixel which has valid phase value in all interferograms and is zero if at least the pixel doesn't have valid phase value in one of the interferograms.

+++++

modifyNetwork.py

To modify the network of interferograms.

+++++

plot.py

To plot the geocoded PySAR product (velocity, temporal coherence, rmse).

+++++

plotNetwork.py

plotting the network of interferograms. And the baseline history.

+++++

save_kml.py

To generate a kml file of the geocoded PySAR products.

+++++

simulation.py

To simulate a stack of interferograms based on a real network of interferograms, and an existing velocity field. Unwrapping error also can be added to the simulation.

+++++

subset.py

Generates a subset of interferograms.

+++++

subtract_add.py

To subtract or add a velocity model to the time-series.

+++++

temporal_coherence.py

generates the temporal coherence.

+++++

timeseries2velocity.py

Generates the velocity field from the time-series.

+++++

timeseries_DEMerror_correction.py

To correct the topographic residuals in the time-series.

+++++

transect.py

To generate the transects of velocity field. If Fault coordinates are introduced then the end point of the profile is modified to force the profile to be perpendicular to the fault.

+++++

tropospheric_correction.py

tropospheric correction using height-correlation approach

+++++

tsviewer.py

time-series viewer

+++++

view.py

A viewer to view all PySAR input files and products (Interferograms, timeseries, velocity, temporal coherence,)

+++++