



Sentinel-1 Interferometry and Unwrapping with GMT5SAR

Credit: ASF Staff

Ubuntu, Python, Python-requests, lxml, GDAL, GMT, GMT5SAR

This recipe creates an interferogram from two Sentinel-1 IW SLCs with VV polarization, and can optionally unwrap it.

In this document you will find:

- A. Background
- B. Materials List
- C. Steps
- D. Sample Granules
- E. Sample Images
- F. Appendix 1: Steps the script completes
- G. Appendix 2: Output files
- H. Appendix 3: Sample script run

A) Background:

In order to perform differential interferometry on Sentinel-1 data using the GMT5SAR software package, ASF provides the python script procS1GMT5SAR.py. This script takes two Sentinel-1 IW SLC files and creates a differential interferogram of the data in both geotiff and KMZ format. The user may specify an input DEM file (which must be in GMT grd format) or a DEM file will automatically be downloaded for the user via the opentopo service. The script creates an interferogram for the main polarization, VV, only.

The Sentinel-1 granules must be in the directory where the script is run, i.e. the .zip files must be in the directory where you run the script. In addition, the ASF provided GMT5SAR configuration file, config.s1a.txt, must be in the same location as your executable file procS1GMT5SAR.py.

B) Materials

- Ubuntu Server 16.04
- At least 200 GB of free space
- Python 2.7 (2.7.12-1ubuntu0~16.04.1)
- Python-requests (2.9.1-3)
- GDAL (gdal-bin (2.1.3+dfsg-1~xenial2)
- GMT (5.2.1+dfsg-4ubuntu2~xenial2)
- GMT5SAR v. 3
- ASF config file: config.s1a.txt
- ASF script: procS1GMT5SAR.py

- Orbit directories (state vectors)
- Sample Granules (Two Sentinel-1 SLCs, IW, VV only)

C) Steps

Preparing the GMT5SAR system:

Starting with a clean installation of Ubuntu Server 16.04 that has at least 200 Gig of free space for data files and the finished product.

Install Software

1. Install python, lxml, and requests:

```
$ sudo apt-get update
$ sudo apt-get install python2.7
$ sudo apt-get install python-lxml
$ sudo apt-get install python-requests
```

2. Install GDAL:

```
$ sudo add-apt-repository -y ppa:ubuntugis/ppa
```

Note: If this first command does not work, you may be using a new version of Ubuntu and the stable GDAL version may not be compatible. To install the latest experimental version of GDAL, follow these commands:

```
$ sudo add-apt-repository -r ppa:ubuntugis/ppa
$ sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
$ sudo apt-get update
```

Experimental versions of GDAL may not behave as expected and may potentially cause other issues with this recipe!

```
$ sudo apt-get update
$ sudo apt-get install gdal-bin libgdal-dev python-gdal
```

3. Install GMT:

```
$ sudo apt-get install csh subversion autoconf libtiff5-dev libhdf5-dev
$ sudo apt-get install ghostscript
$ sudo apt-get install libfftw3-dev
$ sudo apt-get install libgmt-dev
$ sudo apt-get install gmt
```

4. Download and install the orbit directories:

```
$ cd ~  
$ wgethttp://topex.ucsd.edu/gmtsar/tar/ORBITS.tar  
$ cd /usr/local  
$ sudo mkdir orbits  
$ cd orbits  
$ sudo tar -xf ~/ORBITS.tar .
```

5. Download, compile, and install GMT5SAR:

```
$ cd /usr/local  
$ sudo svn checkout svn://gmtserver.soest.hawaii.edu/GMT5SAR/branches/5.4 GMT5SAR  
$ cd GMT5SAR  
$ sudo apt-get install gfortran  
$ sudo autoconf  
$ sudo ./configure --with-orbits-dir=/usr/local/orbits  
$ sudo make  
$ sudo make install
```

Once you have completed the installation steps given above, you are ready to configure.

Version numbers of the packages installed in the creation of this data recipe:

```
python2.7 (2.7.12-1ubuntu0~16.04.1)  
python-lxml (3.5.0-1build1)  
python-requests (2.9.1-3)  
gdal-bin (2.1.3+dfsg-1~xenial2)  
libgdal-dev (2.1.3+dfsg-1~xenial2)  
python-gdal (2.1.3+dfsg-1~xenial2)  
csh (20110502-2.1ubuntu1)  
subversion (1.9.3-2ubuntu1.1)  
autoconf (2.69-9)  
ghostscript (9.18~dfsg~0-0ubuntu2.7)  
libfftw3-dev:amd64 (3.3.4-2ubuntu1)  
libgmt-dev (5.2.1+dfsg-4ubuntu2~xenial2)  
gmt (5.2.1+dfsg-4ubuntu2~xenial2)
```

Configure

Prior to running the ASF script, each user will need to perform these configuration steps:

6. Make GMT5SAR executables locatable:

```
$ export PATH=$PATH:/usr/local/GMT5SAR/bin/
```

Note: You must export the new path every time you login (or set it in your profile).

Download script and config file:

7. Download **config.s1a.txt** from ASF's Data Recipe and place it in your home directory
 - a. The config file is set up to process interferometry only (threshold_snaphu=0.0)
 - b. **For unwrapping**, edit the config file in a text editor, set **threshold_snaphu = 0.2**
8. Download **procS1GMT5SAR.py** from ASF's Data Recipe and place it in your home directory

9. Create a .netrc file:

In order to get the precision state vectors from ASF, you must create a **.netrc** file in your home directory. This file must contain the following information. (Use the username and password from your URS/Earthdata account. Same as used to log-in to Vertex):

Use a text editor such as 'vi' to create or edit the .netrc file:

```
cd ~  
vi .netrc
```

Add these lines:

```
machine urs.earthdata.nasa.gov  
login <username>  
password <password>
```

You are finally ready to create an interferogram!

Run the Script – Create an Interferogram

10. Issue the following command to run the GMT5SAR process:

```
$ python procS1GMT5SAR.py <master granule name> <paired granule name>
```

Notes:

- a) Use two Sentinel-1 SLCs, VV polarization only
- b) The master granule must be first
- c) Use .zip or SAFE files
- d) Script options:
 - a. -h, --help shows help message and exits
 - b. -d DEM, --dem <yourDEMfilename> User specified DEM file (in GMT format). If not used, the script will pull a DEM for you.

Example:

```
$ python procS1GMT5SAR.py  
S1A_IW_SLC_1SSV_20150526T014935_20150526T015002_006086_007E23_679A.zip  
S1A_IW_SLC_1SDV_20150607T014936_20150607T015003_006261_00832E_3626.zip
```

The script will take some time to complete. The three commands prefaced with "Running Interferogram: "takes the bulk of processing time. Expect to wait a significant amount of time for each of the calls to p2p_S1A_TOPS.csh to finish (as much as an hour per call).

D) Sample Granules

S1A_IW_SLC_1SSV_20150526T014935_20150526T015002_006086_007E23_679A.zip

S1A_IW_SLC_1SDV_20150607T014936_20150607T015003_006261_00832E_3626.zip

E) Sample Images

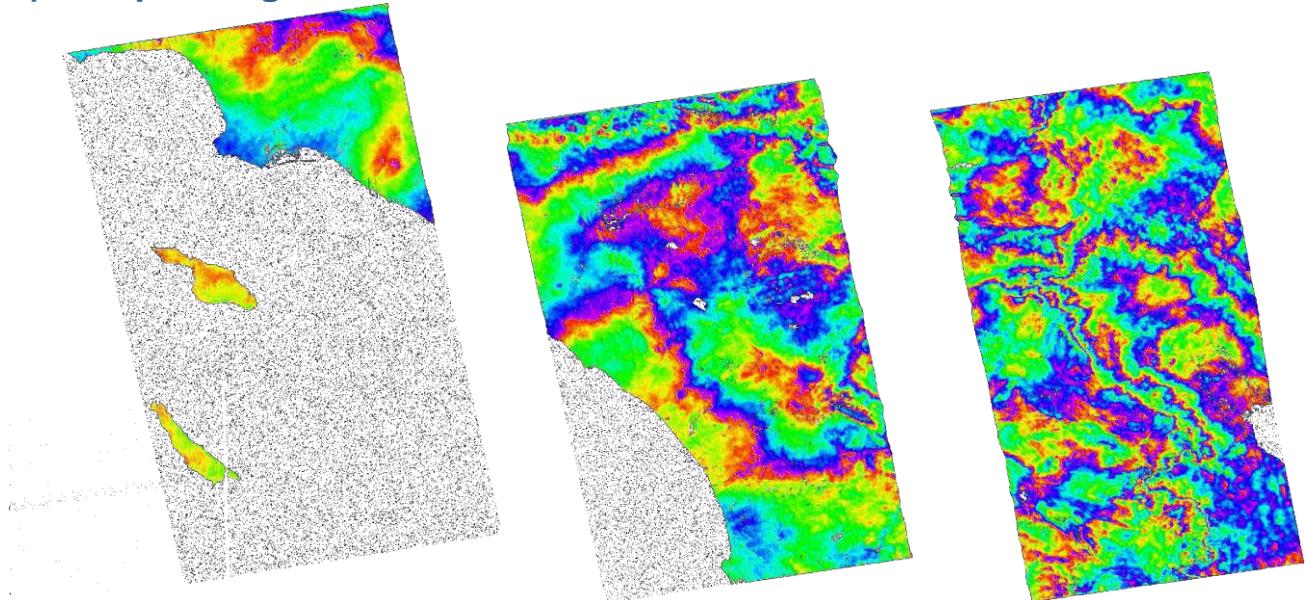


Figure 1. Script output images: color phase products for each swath F1, F2, F3. Contains modified Copernicus Sentinel data (2015) processed by ESA

F) Appendix 1: Steps the Script Completes

How it works:

- 1 Unzip the input Sentinel-1 input granules
- 2 Download appropriate restituted state vectors
- 3 Download DEM covering the area of master image (OpenTopo or user specified)
- 4 Align the two images using the GMT5SAR script align_tops.csh
- 5 Create the interferograms, optionally unwrap, using the GMT5SAR script p2p_S1A_TOPS.csh
- 6 Create and populate the PRODUCT directory

G) Appendix 2: Script Output Files – Unwrapping on

The outputs of the run will be placed in a directory called PRODUCT, e.g. the output from the above run will produce a PRODUCT directory with the following files:

20150526_20150607_F1_amp.tif

20150526_20150607_F1_color_phase.kmz

20150526_20150607_F1_color_phase.png
20150526_20150607_F1_corr.tif
20150526_20150607_F1_unw_phase.kmz
20150526_20150607_F1_unw_phase.png
20150526_20150607_F1_unw_phase.tif
20150526_20150607_F2_amp.tif
20150526_20150607_F2_color_phase.kmz
20150526_20150607_F2_color_phase.png
20150526_20150607_F2_corr.tif
20150526_20150607_F2_unw_phase.kmz
20150526_20150607_F2_unw_phase.png
20150526_20150607_F2_unw_phase.tif
20150526_20150607_F3_amp.tif
20150526_20150607_F3_color_phase.kmz
20150526_20150607_F3_color_phase.png
20150526_20150607_F3_corr.tif
20150526_20150607_F3_unw_phase.kmz
20150526_20150607_F3_unw_phase.png
20150526_20150607_F3_unw_phase.tif

The files are as follows:

F?	- swath designator, F1, F2, or F3
amp	- amplitude image
corr	- coherence map
color_phase	- wrapped phase file
unw_phase	- unwrapped phase file*

With extensions:

.kmz	- Google earth KMZ format
.tif	- Geotiff format
.png	- Portable Network Graphic (browse image)

H) Appendix 3: Sample Script Run

Example Output:

The output from an example run, using the same LA data used in the GMTSAR example found at <http://topex.ucsd.edu/gmtsar/downloads/>, is as follows:

```
> python procS1GMT5SAR.py
S1A_IW_SLC__1SSV_20150526T014935_20150526T015002_006086_007E23_679A.zip
S1A_IW_SLC__1SDV_20150607T014936_20150607T015003_006261_00832E_3626.zip
```

Found master file S1A_IW_SLC__1SSV_20150526T014935_20150526T015002_006086_007E23_679A.zip

Found slave file S1A_IW_SLC__1SDV_20150607T014936_20150607T015003_006261_00832E_3626.zip

Unzipping master file

19 April 2019 v.2.2 | 6

Unzipping slave file
Getting orbit files
--2017-08-09 19:22:32--
https://s1qc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF
Resolving s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)... 137.229.86.187
Connecting to s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)|137.229.86.187|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location:
https://urs.earthdata.nasa.gov/oauth/authorize?client_id=BO_n7nTIIIMljdvU6kRRB3g&redirect_uri=https://vertex.daac.asf.alaska.edu/services/urs4_token_request&response_type=code&state=redirect%3Ahttps%3A%2F%2Fs1qc.asf.alaska.edu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF&app_type=401 [following]
--2017-08-09 19:22:32--
https://urs.earthdata.nasa.gov/oauth/authorize?client_id=BO_n7nTIIIMljdvU6kRRB3g&redirect_uri=https://vertex.daac.asf.alaska.edu/services/urs4_token_request&response_type=code&state=redirect%3Ahttps%3A%2F%2Fs1qc.asf.alaska.edu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF&app_type=401
Resolving urs.earthdata.nasa.gov (urs.earthdata.nasa.gov)... 198.118.243.33, 2001:4d0:241a:4081::89
Connecting to urs.earthdata.nasa.gov (urs.earthdata.nasa.gov)|198.118.243.33|:443... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Reusing existing connection to urs.earthdata.nasa.gov:443.
HTTP request sent, awaiting response... 302 Found
Location:
https://vertex.daac.asf.alaska.edu/services/urs4_token_request?code=2f2aa790b79b3c012b8a61acaf5877cff4f996fadbbc4832e28e524802c99cb6&state=redirect%3Ahttps%3A%2F%2Fs1qc%2Easf%2Ealaska%2Eedu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944%2EEEOF [following]
--2017-08-09 19:22:33--
https://vertex.daac.asf.alaska.edu/services/urs4_token_request?code=2f2aa790b79b3c012b8a61acaf5877cff4f996fadbbc4832e28e524802c99cb6&state=redirect%3Ahttps%3A%2F%2Fs1qc%2Easf%2Ealaska%2Eedu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944%2EEEOF
Resolving vertex.daac.asf.alaska.edu (vertex.daac.asf.alaska.edu)... 137.229.86.197
Connecting to vertex.daac.asf.alaska.edu (vertex.daac.asf.alaska.edu)|137.229.86.197|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location:
https://s1qc.asf.alaska.edu:443/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF [following]
--2017-08-09 19:22:34--
https://s1qc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF
Connecting to s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)|137.229.86.187|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4409878 (4.2M) [text/xml]
Saving to: 'S1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF'

100%[=====

======>] 4,409,878 23.5MB/s in 0.2s

2017-08-09 19:22:35 (23.5 MB/s) -

'S1A_OPER_AUX_POEORB_OPOD_20150615T155109_V20150525T225944_20150527T005944.EOF' saved [4409878/4409878]

--2017-08-09 19:22:35--

https://s1qc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF

Resolving s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)... 137.229.86.187

Connecting to s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)|137.229.86.187|:443... connected.

HTTP request sent, awaiting response... 302 Found

Location:

https://urs.earthdata.nasa.gov/oauth/authorize?client_id=BO_n7nTIIIMljdvU6kRRB3g&redirect_uri=https://vertex.daac.asf.alaska.edu/services/urs4_token_request&response_type=code&state=redirect%3Ahttps%3A%2F%2Fs1qc.asf.alaska.edu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF&app_type=401 [following]

--2017-08-09 19:22:35--

https://urs.earthdata.nasa.gov/oauth/authorize?client_id=BO_n7nTIIIMljdvU6kRRB3g&redirect_uri=https://vertex.daac.asf.alaska.edu/services/urs4_token_request&response_type=code&state=redirect%3Ahttps%3A%2F%2Fs1qc.asf.alaska.edu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF&app_type=401

Resolving urs.earthdata.nasa.gov (urs.earthdata.nasa.gov)... 198.118.243.33, 2001:4d0:241a:4081::89

Connecting to urs.earthdata.nasa.gov (urs.earthdata.nasa.gov)|198.118.243.33|:443... connected.

HTTP request sent, awaiting response... 401 Unauthorized

Reusing existing connection to urs.earthdata.nasa.gov:443.

HTTP request sent, awaiting response... 302 Found

Location:

https://vertex.daac.asf.alaska.edu/services/urs4_token_request?code=a425247b9ca1ea75bebbb622ba3fc0fa9fe950f5aacfa3b6d57e7def61b0760&state=redirect%3Ahttps%3A%2F%2Fs1qc%2Easf%2Ealaska%2Eedu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944%2EEEOF [following]

--2017-08-09 19:22:36--

https://vertex.daac.asf.alaska.edu/services/urs4_token_request?code=a425247b9ca1ea75bebbb622ba3fc0fa9fe950f5aacfa3b6d57e7def61b0760&state=redirect%3Ahttps%3A%2F%2Fs1qc%2Easf%2Ealaska%2Eedu%3A443%2Faux_poeorb%2FS1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944%2EEEOF

Resolving vertex.daac.asf.alaska.edu (vertex.daac.asf.alaska.edu)... 137.229.86.197

Connecting to vertex.daac.asf.alaska.edu (vertex.daac.asf.alaska.edu)|137.229.86.197|:443... connected.

HTTP request sent, awaiting response... 302 Found

Location:

https://s1qc.asf.alaska.edu:443/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF [following]

--2017-08-09 19:22:37--

https://s1qc.asf.alaska.edu/aux_poeorb/S1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF

Connecting to s1qc.asf.alaska.edu (s1qc.asf.alaska.edu)|137.229.86.187|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 4420094 (4.2M) [text/xml]
Saving to: 'S1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF'

100%[=====>] 4,420,094 --K/s in 0.1s

2017-08-09 19:22:37 (34.8 MB/s) -
'S1A_OPER_AUX_POEORB_OPOD_20150627T155155_V20150606T225944_20150608T005944.EOF' saved
[4420094/4420094]

Getting DEM file

```
wget -Odem.tif "http://opentopo.sdsc.edu/otr/getdem?demtype=SRTMGL1&west=-119.008456807&south=32.3644581974&east=-115.701091959&north=34.6778684913&outputFormat=GTiff"  
--2017-08-09 19:22:37-- http://opentopo.sdsc.edu/otr/getdem?demtype=SRTMGL1&west=-119.008456807&south=32.3644581974&east=-115.701091959&north=34.6778684913&outputFormat=GTiff  
Resolving opentopo.sdsc.edu (opentopo.sdsc.edu)... 198.202.90.222  
Connecting to opentopo.sdsc.edu (opentopo.sdsc.edu)|198.202.90.222|:80... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: http://ot-data1.sdsc.edu:9090/otr/getdem?demtype=SRTMGL1&west=-119.008456807&south=32.3644581974&east=-115.701091959&north=34.6778684913&outputFormat=GTiff  
[following]  
--2017-08-09 19:22:38-- http://ot-data1.sdsc.edu:9090/otr/getdem?demtype=SRTMGL1&west=-119.008456807&south=32.3644581974&east=-115.701091959&north=34.6778684913&outputFormat=GTiff  
Resolving ot-data1.sdsc.edu (ot-data1.sdsc.edu)... 198.202.90.171  
Connecting to ot-data1.sdsc.edu (ot-data1.sdsc.edu)|198.202.90.171|:9090... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [application/octet-stream]  
Saving to: 'dem.tif'
```

[<=>] 198,390,002 38.8MB/s in
6.3s

2017-08-09 19:22:54 (30.1 MB/s) - 'dem.tif' saved [198390002]

S1A20150526_014935_F1

S1A20150607_014936_F1

Writing 31 lines for orbit...

Writing 31 lines for orbit...

Writing 52 lines of precise orbit for the LED file...

Writing 52 lines of precise orbit for the LED file...

Successfully opened S1A20150526_014935_F1.PRM

Successfully opened S1A20150607_014936_F1.PRM

Elapsed time 00:00:19.036 | (grdfilter) |

[1] 1551

[2] 1552

[2] + Done SAT_llt2rat S1A20150607_014936_F1.PRM 1 < topo.llt > slave.ratll

[1] + Done SAT_llt2rat S1A20150526_014935_F1.PRM 1 < topo.llt > master.ratll

[1] 1591

[2] 1592

```

[2] + Done      gmt surface atmp.xyz -bi3d -R0/20664/0/12196 -l16/8 -T0.3 -Gatmp.grd -N1000 -r
[1] + Done      gmt surface rtmp.xyz -bi3d -R0/20664/0/12196 -l16/8 -T0.3 -Grtmp.grd -N1000 -r
Writing 31 lines for orbit...
Writing SLC..Image Size: 21928 X 12196...
Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9
number of points clipped to short int 14
Writing 31 lines for orbit...
Reading in range and azimuth shifts table...
Writing SLC..Image Size: 20664 X 12196...
Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9
number of points clipped to short int 47
Writing 52 lines of precise orbit for the LED file...
Writing 52 lines of precise orbit for the LED file...
Successfully opened S1A20150526_014935_F1.PRM
Successfully opened S1A20150607_014936_F1.PRM
S1A20150526_014936_F2
S1A20150607_014936_F2
Writing 31 lines for orbit...
Writing 31 lines for orbit...
Writing 52 lines of precise orbit for the LED file...
Writing 52 lines of precise orbit for the LED file...
Successfully opened S1A20150526_014936_F2.PRM
Successfully opened S1A20150607_014936_F2.PRM
Elapsed time 00:00:17.434 | (grdfilter) |
[1] 2221
[2] 2222
[2] Done      SAT_llt2rat S1A20150607_014936_F2.PRM 1 < topo.llt > slave.ratll
[1] + Done     SAT_llt2rat S1A20150526_014936_F2.PRM 1 < topo.llt > master.ratll
[1] 2235
[2] 2236
[2] Done      gmt surface atmp.xyz -bi3d -R0/24640/0/12196 -l16/8 -T0.3 -Gatmp.grd -N1000 -r
[1] + Done     gmt surface rtmp.xyz -bi3d -R0/24640/0/12196 -l16/8 -T0.3 -Grtmp.grd -N1000 -r
Writing 31 lines for orbit...
Writing SLC..Image Size: 25788 X 12192...
Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9
number of points clipped to short int 6
Writing 31 lines for orbit...
Reading in range and azimuth shifts table...
Writing SLC..Image Size: 24640 X 12196...
Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9
number of points clipped to short int 9
Writing 52 lines of precise orbit for the LED file...
Writing 52 lines of precise orbit for the LED file...
Successfully opened S1A20150526_014936_F2.PRM
Successfully opened S1A20150607_014936_F2.PRM
S1A20150526_014937_F3
S1A20150607_014937_F3
Writing 31 lines for orbit...

```

19 April 2019 v.2.2 | 10

Writing 31 lines for orbit...

Writing 52 lines of precise orbit for the LED file...

Writing 52 lines of precise orbit for the LED file...

Successfully opened S1A20150526_014937_F3.PRM

Successfully opened S1A20150607_014937_F3.PRM

Elapsed time 00:00:17.434 | (grdfilter) |

[1] 3446

[2] 3447

[2] Done SAT_llt2rat S1A20150607_014937_F3.PRM 1 < topo.llt > slave.ratll

[1] + Done SAT_llt2rat S1A20150526_014937_F3.PRM 1 < topo.llt > master.ratll

[1] 3727

[2] 3728

[2] Done gmt surface atmp.xyz -bi3d -R0/23848/0/12196 -I16/8 -T0.3 -Gatmp.grd -N1000 -r

[1] + Done gmt surface rtmp.xyz -bi3d -R0/23848/0/12196 -I16/8 -T0.3 -Grtmp.grd -N1000 -r

Writing 31 lines for orbit...

Writing SLC..Image Size: 24864 X 12196...

Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9

number of points clipped to short int 0

Writing 31 lines for orbit...

Reading in range and azimuth shifts table...

Writing SLC..Image Size: 23848 X 12196...

Working on burst #1 #2 #3 #4 #5 #6 #7 #8 #9

number of points clipped to short int 1

Writing 52 lines of precise orbit for the LED file...

Writing 52 lines of precise orbit for the LED file...

Successfully opened S1A20150526_014937_F3.PRM

Successfully opened S1A20150607_014937_F3.PRM

Running Interferogram: p2p_S1A_TOPS.csh S1A20150526_014935_F1 S1A20150607_014936_F1 config.s1a.txt
> log 2>&1

Running Interferogram: p2p_S1A_TOPS.csh S1A20150526_014936_F2 S1A20150607_014936_F2 config.s1a.txt
> log 2>&1

Running Interferogram: p2p_S1A_TOPS.csh S1A20150526_014937_F3 S1A20150607_014937_F3 config.s1a.txt
> log 2>&1

Changing name field in KML file (phasefilt_mask_ll.kml)

Writing output file (20150526_20150607_F1_vv_phase.kml)

Changing name field in KML file (unwrap_mask_ll.kml) Writing output file (20150526_20150607_F1_vv_unwrap.kml)

Changing name field in KML file (phasefilt_mask_ll.kml)

Writing output file (20150526_20150607_F2_vv_phase.kml)

Changing name field in KML file (unwrap_mask_ll.kml) Writing output file (20150526_20150607_F2_vv_unwrap.kml)

Changing name field in KML file (phasefilt_mask_ll.kml)

Writing output file (20150526_20150607_F3_vv_phase.kml)

Changing name field in KML file (unwrap_mask_ll.kml) Writing output file (20150526_20150607_F3_vv_unwrap.kml)