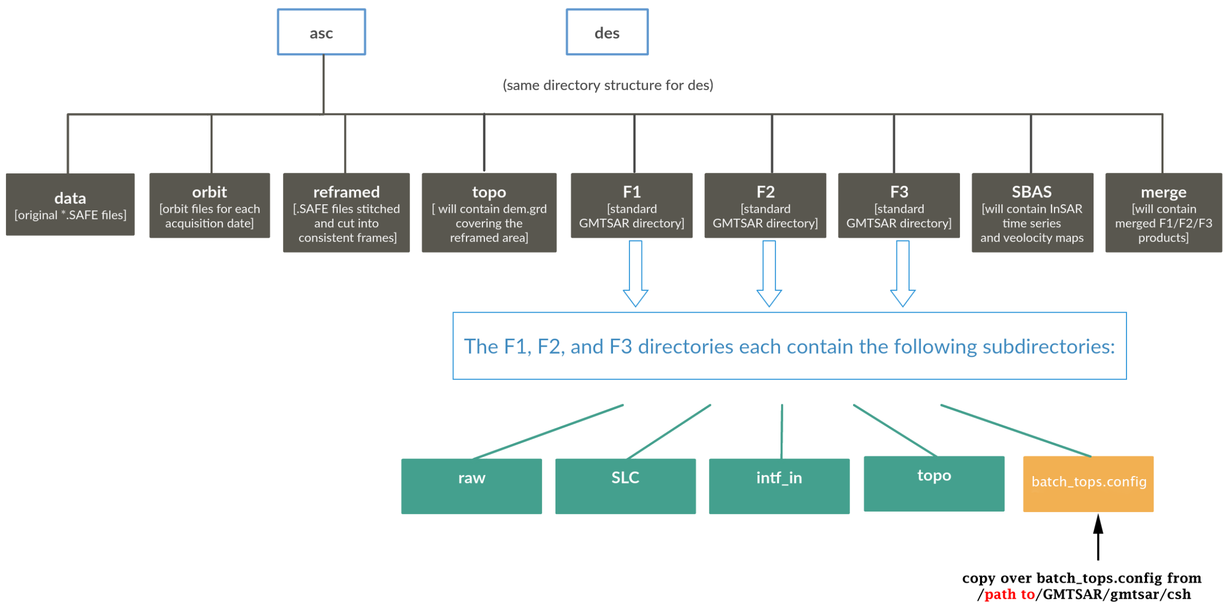
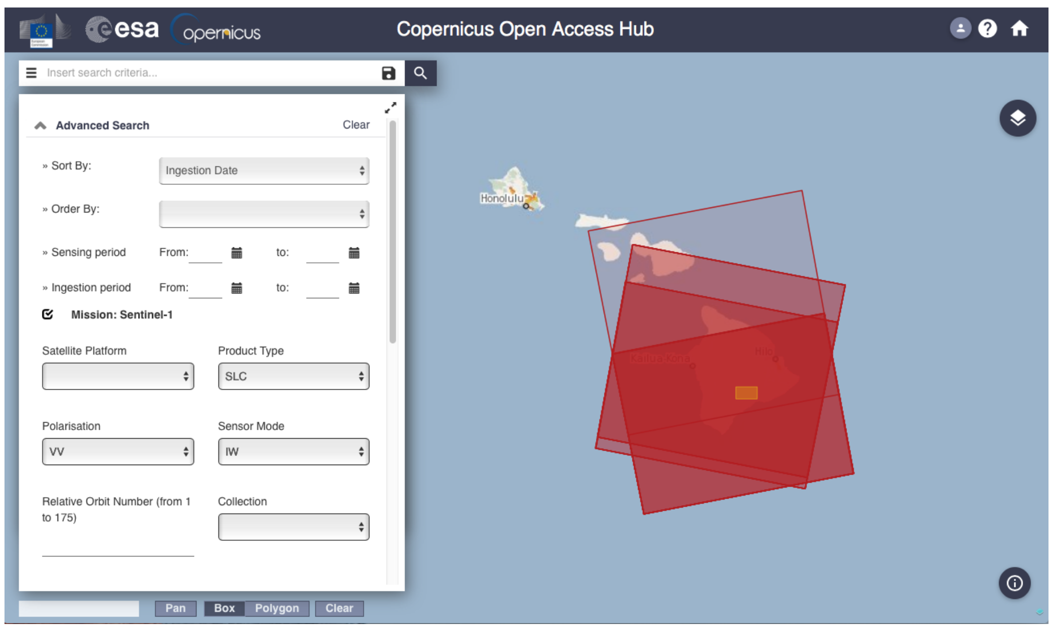
How to make an InSAR time series from Sentinel-1 TOPS data: Kilauea

1. Make a top level **directory structure** for ascending or descending data as follows:

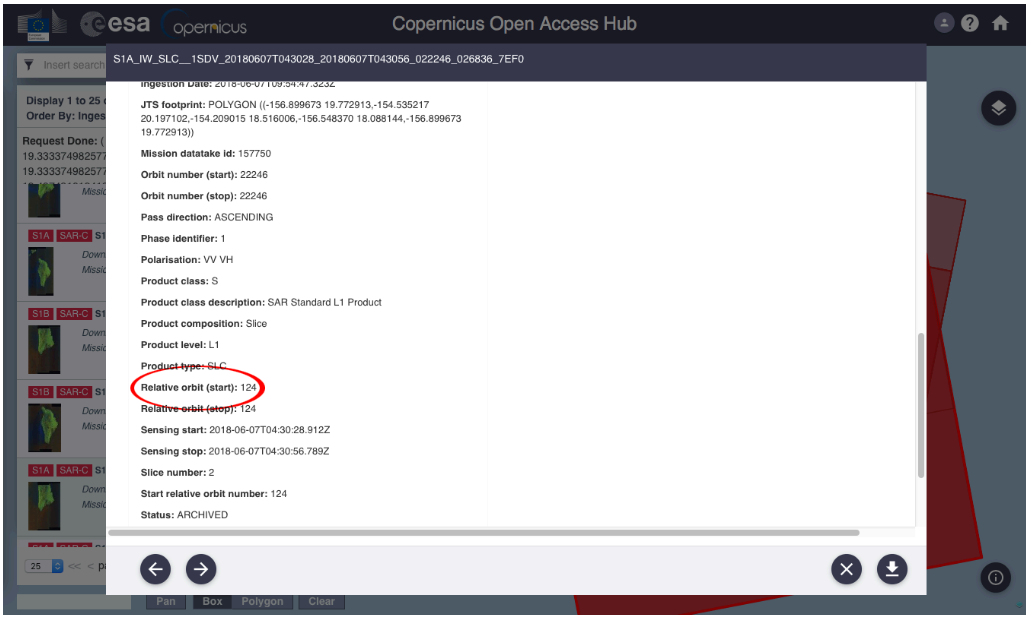


2. Prepare a **topography grid** (dem.grd)
   a) Using the DEM generator web interface (http://topex.ucsd.edu/gmtsar/demgen/), prepare a topography grid for your area of interest (lon -157.0 to -154.2, lat 18.0 to 20.4 for Kilauea). This will download as dem.grd, along with a kml companion file. View the kml file in Google Earth and verify that it completely covers the area (see below).
   b) Move the dem.grd to your upper level **topo** directory (inside **asc** or **des** directories).
   c) Next link the dem.grd file to a **topo** directory inside the F1/F2/F3 directories.. Create a symbolic link in each:
      cd asc/F1/
      mkdir topo
      cd topo
      ln -s ../../topo/dem.grd .
   d) Also link the dem.grd to the **merge** directory (doesn't need to be inside a topo folder here).
      cd asc/merge/
      ln -s ../../topo/dem.grd .

3. Plan your overall **processing strategy**
    a. Go to the Sentinel open Copernicus data hub GUI, login or make an account, and search for data: (https://scihub.copernicus.eu/dhus/#/home)
    b. Use the Pan and Box tools (bottom left corner) to zoom in on the area of interest, for example, The Big Island, Hawaii (lon -157.0 to -154.2, lat 18.0 to 20.4, an area that spans the entire frame for making the topography). Draw a box over this area.
    c. Using the drop down search menu (three horizontal lines icon), set the search parameters for the sensing dates, product type (SLC), polarisation (VV) and the sensor mode (IW). As you can see in the example below, the search returned both ascending and descending frames. Ascending frames are oriented NW/SE while descending frames are oriented NE/SW.



    d. From the search results, select one descending frame and one ascending frame and take note of the relative orbit numbers of each (in this case, 124 for ascending and 87 for descending). This is found if you click on the eyeball icon on the individual scene (left hand side of the screen) and click on the "product" drop-down menu (see screenshot below). You'll want to download data from ascending and descending tracks separately, since the data files for each will go into the different **asc** and **des** directories you created before. All instructions below are for downloading and processing ascending data specifically, but they can be replicated for descending data.



4. Download Data (2 Options)
    a. To **download directly** from the Copernicus hub, do the data search again but this time include in your search one of the relative orbit numbers that you just found (for either ascending or descending), along with the same parameters as before (desired sensing dates, polarisation= VV (HH for Antarctica), product type = SLC, sensor mode = IW). Then manually click on each result and download. Put these .zips into the corresponding data directory that you created earlier (**asc/data** or **des/data**).

    OR

    b. **Automated download** method using ASF API (Alaska Satellite Facility)
        i. If you are new user, you will need to set up a user account and accept the license agreement.
        ii. Generate URL command with web API Query (https://www.asf.alaska.edu/get-data/learn-by-doing/) and use the wget command to download the results (see example scripts below for Kilauea data).

        Sample scripts for Kilauea data:
        - You will need to enter your own intended path to **data** storage location (**asc/data**) and your own username and password (highlighted in the scripts)

```
#!/bin/bash
#
# search the ASF site to determine all data inside of a polygon and make a csv file listing file names
#
```

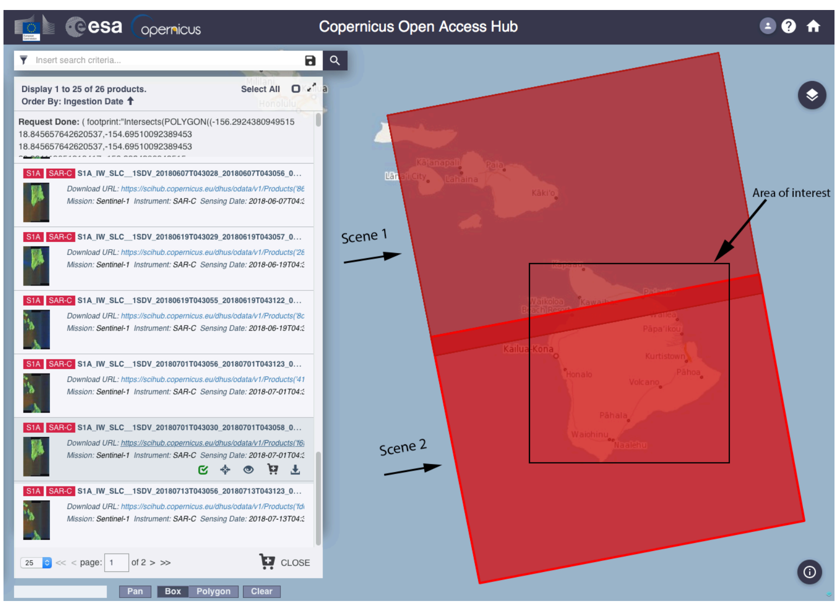
```
cd  path to /asc/data
cmd="https://api.daac.asf.alaska.edu/services/search/param?platform=Sentinel-1A,Sentinel-1B&polygon=-
        155.64625964,19.3937790141,-155.52834329,19.1922612441,-154.994620338,19.4236985176,-
        155.135252149,19.645334611,-
        155.64625964,19.3937790141&processingLevel=SLC&relativeOrbit=124&output=csv"
#
echo $cmd | xargs curl  > asf.csv
#
cat asf.csv | awk -F"," '{if (NR>1) print $27}' | awk -F'"' '{print $2}' > data.list
```

```
#!/bin/csh
#
#  download the scenes from the data.list file that was created by above script
#
cd  path to /asc/data
foreach file (`awk '{print $1}' data.list`)
 set name = `echo $file | awk -F'/' '{print $6}' | awk -F'.' '{print $1".SAFE"}'`
 if ( ! -e $name ) then
   echo $name
   wget --http-user=**** --http-password=***** $file
 endif
end
```

---

iii. Unzip all files and remove the .zip files. Note, all Sentinel-1 directories have a **.SAFE** suffix

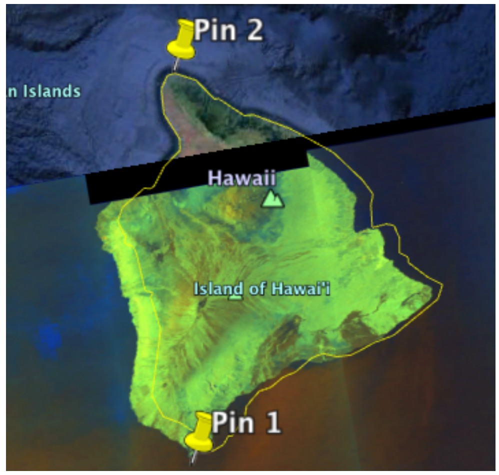
5. **Defining region of interest**: Stitching multiple frames and/or cropping one frame to define a smaller region.



a. You may need to **stitch frames** together because the area of interest spans two frames. If you have *.SAFE file with names that share a common date and must be stitched, stitch them by using pins in Google Earth.

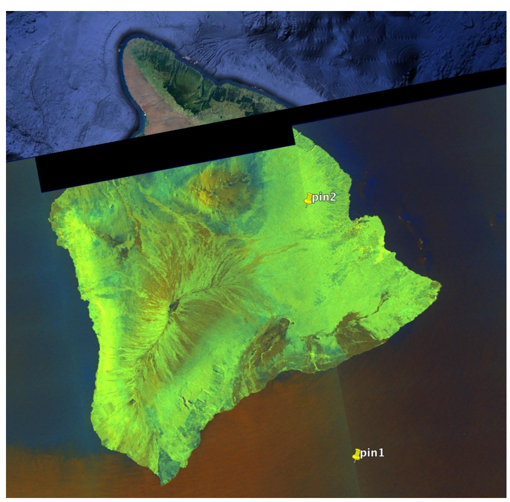
   i. Open the kml preview found in **data**/*.SAFE/preview/map-overlay.kml

ii. Drop two pins ordered in the **along-track direction.** Note that this will differ for **asc** vs. **des** directories. The example photo below is the pin order for ascending, but a descending orbit would require swapping pins 1 and 2.



iii. If you are only interested in a small area, the pins can be placed within one frame instead (see example below), and the regions above and below these constraints will be cut off. This is helpful for making processing run more quickly.

In this case, pin 1 could be somewhere south of Kilauea (ocean), and pin 2 could be on land, in the ascending orbit path direction (see below). For descending, pin 1 could be somewhere due-North of Hilo (ocean), and pin 2 could be somewhere on land near Kilauea.



b. Export coordinates of the pins to a file called pins.ll and save this file in the **reframed** directory.

Example file coordinates for ascending might be:

-155.13 18.95
-155.30 19.75

Example file coordinates for descending might be:

-155.50 19.65
-155.680 18.93

c. Organize files having **precise** orbit files.
   i. Create a file in the **data** directory called "SAFE_filelist", and input a list of full paths to all of your .SAFE files in order of acquisition date:

```
/ full path to /asc/data/S1A_IW_SLC__1SDV_20180219T043025_20180219T043053_020671_023679_5817.SAFE
/ full path to /asc/data/S1A_IW_SLC__1SDV_20180315T043025_20180315T043053_021021_024191_5F9B.SAFE
/ full path to /asc/data/S1A_IW_SLC__1SDV_20180327T043025_20180327T043053_021196_024722_A9DB.SAFE
. . .
```

   ii. The next step requires use of the organize_files_tops.csh script, which searches for and downloads the appropriate **precise** orbit files for you. To use this script, add the following to your .cshrc or .tcshrc file:

```
alias wgetasf 'wget --http-user=**** --http-password=*****'
```

   where the username and password are those for your ASF account. Remember to close and open a new terminal after doing so.

   iii. Run the script in two steps in **data** directory.
   - MAC USERS:

     First run with mode 1:
     `organize_files_tops.csh SAFE_filelist pins.ll 1`
     Then with mode 2:
     `organize_files_tops.csh SAFE_filelist pins.ll 2`

   - LINUX USERS:

     First run with mode 1:
     `organize_files_tops_linux.csh SAFE_filelist pins.ll 1`
     Then with mode 2:
     `organize_files_tops_linux.csh SAFE_filelist pins.ll 2`

   iv. Now a new directory within the **data** directory named Fxxxx_Fxxxx has been created and populated with new stitched/trimmed *SAFE data.

d. Organize files with **restituted** orbits (optional step):
   Data that is from within the past 20 days will not yet have precise orbit files available. If you want to use newer data, then you need to download the appropriate restituted (RESORB) orbit files and use the **create_frame_tops.csh** script instead.
   i. Using either ESA (https://qc.sentinel1.eo.esa.int/aux_resorb/) or ASF (https://s1qc.asf.alaska.edu/aux_resorb/), download the appropriate orbit files that correspond to each .SAFE. Note that ESA orbit files will

likely be shifted by 1 day compared to ASF's orbit files, so we recommend that ASF orbits be used.

Orbit files must span the acquisition date and time. For example, for SAR data file

S1A_IW_SLC__1SDV_20180514T043027_20180514T043055_021896_025D31_C20C.SAFE ,

Download orbit file

S1A_OPER_AUX_RESORB_OPOD_20180514T065858_V20180514T024649_20180514T06041
9.EOF

ii. Do this for each *.SAFE you have that requires a restituted orbit file and place these orbit files (*.EOF) in the **asc** or **des orbit** directory .

iii. cd to the **reframed** directory and run the commands below for each *.SAFE and its corresponding *.EOF:

```
ln -s /full-path-to/asc/orbit/S1A_OPER_AUX_RESORB_OPOD_20180514T065858_V20180514T024649_20180514T060419.EOF .

create_frame_tops.csh  SAFE_list  S1B_OPER_AUX_POEORB_OPOD_20180609T110546_V20180519T225942_20180521T005942.EOF
```

iv. Move the resulting *.SAFE file into the same Fxxxx_Fxxxx folder (see above step c) where the rest of the precise *.SAFE data was placed.

v. Repeat steps iii and iv for each *.SAFE you have that requires a restituted orbit file.

6. **Align** all the slaves to the master

a. Go to the **F1/raw** directory. Link the data and orbit files that belong to this (F1) subswath. Also link the dem.grd.

```
ln -s ../../data/F*/*.SAFE/*/*iw1*vv*xml .
ln -s ../../data/F*/*.SAFE/*/*iw1*vv*tiff .
ln -s ../../data/*EOF .
ln -s ../topo/dem.grd .
```

Note: "iw1" files correspond to the F1 subswath, "iw2" files correspond to the F2 subswath, etc. When you repeat this step for the **F2**/**F3** directories, change the 1's to 2's or 3's!

b. Now you will need to prepare a file called **data.in** that has the data file names (no suffix) and orbit file names organized as follows:

```
s1a-iw1-slc-vv-20180207t043037-20180207t043048-020496-0230e3-005:S1A_OPER_AUX_POEORB_OPOD_20180227T120553_V20180206T225942_20180208T005942.EOF
s1a-iw1-slc-vv-20180219t043037-20180219t043048-020671-023679-005:S1A_OPER_AUX_POEORB_OPOD_20180311T120551_V20180218T225942_20180220T005942.EOF
s1a-iw1-slc-vv-20180303t043037-20180303t043048-020846-023c05-005:S1A_OPER_AUX_POEORB_OPOD_20180323T120847_V20180302T225942_20180304T005942.EOF
s1a-iw1-slc-vv-20180315t043037-20180315t043048-021021-024191-005:S1A_OPER_AUX_POEORB_OPOD_20180404T120811_V20180314T225942_20180316T005942.EOF
s1a-iw2-slc-vv-20180327t043038-20180327t043049-021196-024722-005:S1A_OPER_AUX_POEORB_OPOD_20180416T120736_V20180326T225942_20180328T005942.EOF
s1a-iw2-slc-vv-20180408t043038-20180408t043049-021371-024c9b-005:S1A_OPER_AUX_POEORB_OPOD_20180428T120749_V20180407T225942_20180409T005942.EOF
s1a-iw2-slc-vv-20180420t043038-20180420t043049-021546-025211-005:S1A_OPER_AUX_POEORB_OPOD_20180510T120714_V20180419T225942_20180421T005942.EOF
s1a-iw2-slc-vv-20180502t043039-20180502t043050-021721-025793-005:S1A_OPER_AUX_POEORB_OPOD_20180522T120730_V20180501T225942_20180503T005942.EOF
s1b-iw1-slc-vv-20180508t042957-20180508t043008-010825-013ca6-005:S1B_OPER_AUX_POEORB_OPOD_20180528T110541_V20180507T225942_20180509T005942.EOF
s1a-iw2-slc-vv-20180514t043040-20180514t043051-021896-025d31-005:S1A_OPER_AUX_POEORB_OPOD_20180603T120720_V20180513T225942_20180515T005942.EOF
                    ...
```

i. There is a script **prep_data.csh (mac) or prep_data_linux.csh (linux)** that will do this for you but only for the precise (not restituted) orbits. Simply run the command "prep_data.csh" from the **raw** directory. After running the script, open and manually provide the name of any missing restituted orbits (if you are using them).

c. Next preprocess and align the images specified in data.in

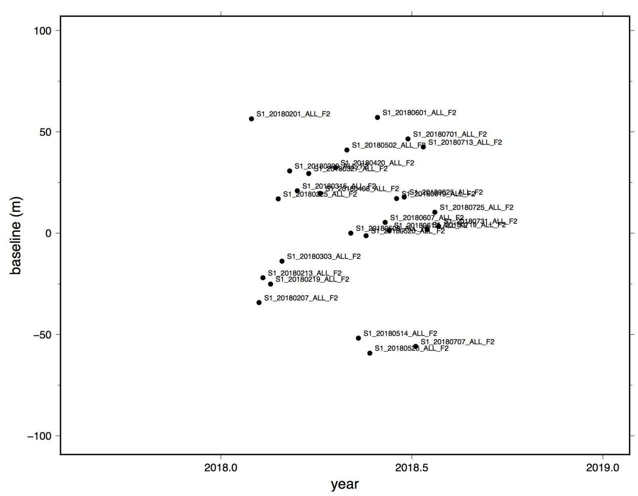
   i. From the **raw** directory, run preproc_batch_tops.csh with mode = 1. This generates a table of baselines (baseline_table.dat):

      preproc_batch_tops.csh data.in dem.grd 1

   ii. Save (move) baseline_table.dat for later use:

      mv baseline_table.dat ../

   iii. Check on the baseline.ps (figure showing baselines) to determine a master for the stack of data. Select an image in the middle. For the case below, image 0505 is a good master.



   iv. The master you choose needs to get promoted to the **first line** in **data.in**. Open/edit this file and move (copy/delete/paste) the line designating the master (in this case, 20180508) to the first line of data.in, so the program will treat it as the master.

d. Re-run preproc_batch_tops.csh with mode = 2. This will take some time (~hour-ish?)

      preproc_batch_tops.csh data.in dem.grd 2

e. Repeat steps a-d for directories **F2** and **F3**, changing iw1 to iw2/iw3 in the links of step a.

7. Make a list of all the **interferograms** with temporal and baseline constraints.

   a. Go to F1directory and run

      select_pairs.csh baseline_table.dat 50 100

Here, 50 is the temporal baseline (days) and 100 is the perpendicular baseline (meters). The output file **intf.in** then contains interferometric pairs with a timespan smaller than 50 days and a perpendicular baseline smaller than 100 meters. The contents of **intf.in** look as follows:

```
S1_20180508_ALL_F1:S1_20180514_ALL_F1
S1_20180508_ALL_F1:S1_20180526_ALL_F1
S1_20180508_ALL_F1:S1_20180607_ALL_F1
S1_20180508_ALL_F1:S1_20180520_ALL_F1
…
```

- To view their baselines, open up baseline.ps. To obtain a count of the interferogram pairs specified in intf.in, use wc -l intf.in

b. Repeat step a for the **F2** and **F3** directories.

c. Copy the intf.in file to the **F2** and **F3** directories. Copy the batch_tops.config file from **F1** into the **F2** and **F3** directories.

8. Now it's time to make some **interferograms**.

For this Kilauea example specifically, subswath F1 happens to contain only ocean. Running the interferograms for F1 therefore isn't necessary, but it's good practice and keeps the steps consistent between subswath directories. Two ways of checking if you'll need all 3 subswaths are by 1) reviewing a quick-look png file in the original data folder, or 2) checking the topo_ra.pdf file after running step 8b in each **F1/2/3** directory - if it is just noise, you can skip processing for that subswath.

a. In the batch_tops.config file in **F1**, set some parameters:

   i. Set the master_image (example below, use yours though)

```
master_image = S1_20180508_ALL_F1
```

   Set/check the following:

```
Proc_stage = 1
shift_topo = 0
filter_wavelength = 200
range_dec = 8
azimuth_dec = 2
threshold_snaphu = 0
threshold_geocode = 0
```

   This will decimate the image 8 in range and 2 in azimuth. Keep these numbers consistent across sub-swaths (i.e., when you do this for the F2 and F3 directories). The two zeros will skip unwrapping and geocoding as this will be done after merging the subswaths.

b. Next we need to map topography into phase to be used by the master so edit batch_tops.config by setting proc_stage = 1 and make a single interferogram. This makes a large trans.dat file, as well as a topo_ra.pdf file that you can check.

head -1 intf.in > one.in
intf_tops.csh one.in batch_tops.config

c. Now edit batch_tops.config again and set proc_stage = 2.

d. Now generate the interferograms specified in **intf.in**. (This takes a while)
   i. Run this from **F1**:

      intf_tops_parallel.csh intf.in batch_tops.config 6

      - To use this script, you'll need to have parallel installed. If it's not, first run the command "sudo port install parallel", or just use the intf_tops.csh script instead and remove the 6 from the end of the command. '6' is the number of threads that are being sent out, if you are working on a larger server this can be increased (for each thread, allow one core and 5-8 GB of memory).
      - Output interferograms will be put in a folder called intf_all

e. Repeat steps a-e for directories **F2** and **F3**.
   i. Before doing so, make sure to edit intf.in in each **F2/3** directory, replacing all F1 the entries with F2 or F3.
      - A handy way to do this in vi is the following:

        vi intf.in
        :%s/F1/F2/g (for subswath F3, change F2 to F3)

   ii. Also edit the batch_tops.config file. Change the F1 designation in the master_image parameter to F2 or F3.

       master_image = S1_20180508_ALL_F2

   Check that the other parameters are set correctly and that proc_stage = 1 for the first interferogram, and then modified to 2 for the full set.

9. **Merge** subswaths

   a. Go to the **merge** folder and prepare a merge_list for merge_batch.csh as follows:

IF1_Swath1_Path:master.PRM:repeat.PRM,IF1_Swath2_Path:master.PRM:repeat.PRM,IF1_Swath3_Path:master.PRM:repeat.PRM

../F2/intf_all/2018127_2018133/:S1A20180508_ALL_F2.PRM:S1A20180514_ALL_F2.PRM,../F3/intf_all/2018127_2018133/:S1A20180508_ALL_F3.PRM:S1A20180514_ALL_F3.PRM
../F2/intf_all/2018127_2018139/:S1A20180508_ALL_F2.PRM:S1A20180520_ALL_F2.PRM,../F3/intf_all/2018127_2018139/:S1A20180508_ALL_F3.PRM:S1A20180520_ALL_F3.PRM
../F2/intf_all/2018127_2018145/:S1A20180508_ALL_F2.PRM:S1A20180526_ALL_F2.PRM,../F3/intf_all/2018127_2018145/:S1A20180508_ALL_F3.PRM:S1A20180526_ALL_F3.PRM

   Place a line which includes the super master as the first line in the merge_list.

   b. Copy the batch_tops.config to the merge directory. Make sure threshold_snahpu = 0 to skip unwrapping (we'll unwrap later). Add dem link too.

ln -s ../topo/dem.grd .

c. Run **merge_batch.csh** from the merge directory:
merge_batch.csh merge_list batch_tops.config
The output will be interferometric folders that contain merged coherence, mask, and phase files.

10. **Unwrap** the phase from the **merge** directory:
   a. Unwrap the phase for each interferogram (unless your area will be difficult to unwrap due to vegetation/low coherence, in which case see part b).
      Here is sample script that does this (make an intflist with a list of interferogram directories to feed to your script). Name this script "unwrap_intf.csh":

```
____________________________________
#!/bin/csh -f
# intflist contains a list of all date1_date2 directories.

foreach line (`awk '{print $1}' intflist`)
 cd $line
 snaphu_interp.csh 0.10
 cd ..
end
____________________________________
```

      - Optionally add region cut to script: snaphu_interp.csh [unwrap threshold] [smoothing factor] [region cut range_min/range_max/az_min/az_max]

      There is also an option to instead run this in parallel with the unwrap_parallel.csh script. This calls the script you just created (unwrap_intf.csh), so make sure you named it correctly. Again, 6 is the number of threads.

      unwrap_parallel.csh intflist 6

   b. User option: If you are using data from an area that will be tricky to unwrap (i.e. vegetated region such as Hawaii), there are some steps below for stacking the coherence and creating a mask first, then unwrapping.
      i. Stack the coherence: Create a corr_stack.grd file which is an average of a selection of coherence files (corr.grd) you put in.

```
gmt grdmath 2006355_2007036/corr.grd 2006355_2009179/corr.grd
    2006355_2009225/corr.grd 2007036_2009225/corr.grd
    2008039_2010136/corr.grd ADD ADD ADD ADD 5 DIV = corr_stack.grd
```

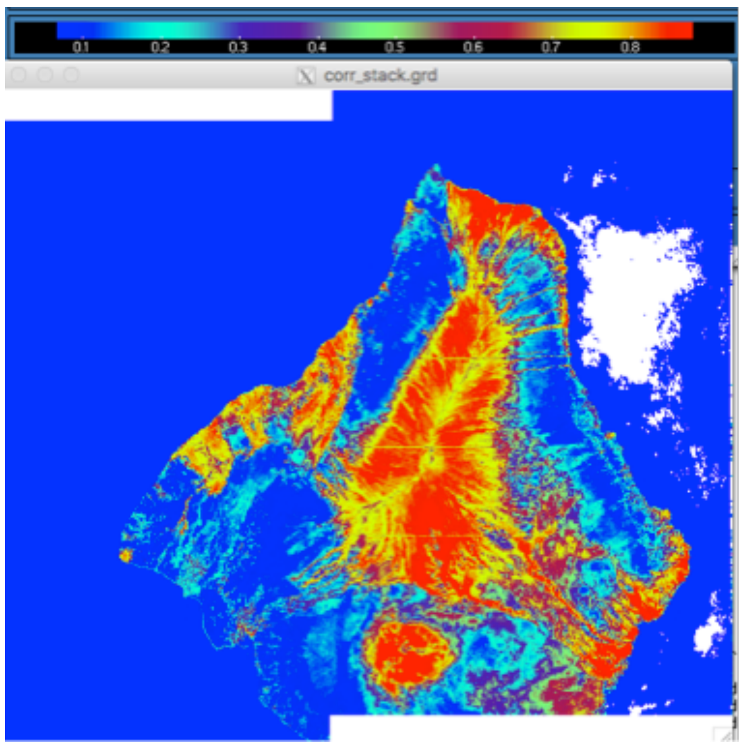
Alternatively, you can save some time by instead creating corr_stack.grd with the **stack.csh** script. Run the following from the **merge** directory:

```
ls 201*/corr.grd > corr.grd_list
stack.csh corr.grd_list 1 corr_stack.grd std.grd
```

"1" is the scaling factor (in this case, no scaling). "corr_stack.grd" and "std.grd" are the output files (we are only going to use "corr_stack.grd").

After creating corr_stack.grd, take a look at it:

```
ncview corr_stack.grd
```

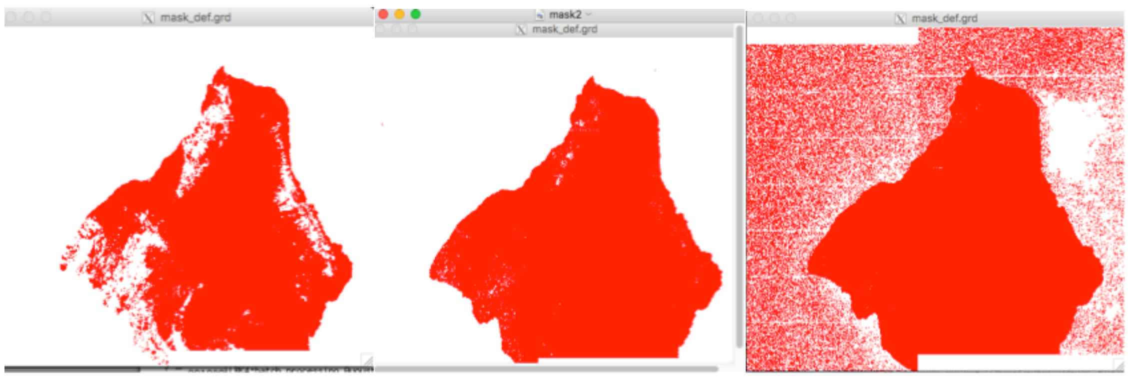


Here we can see that most of the old lava flows show high coherence, while the more vegetated regions are low.

ii. Now we'll create an ocean mask based on the stacked coherence. This will mask out any data that doesn't meet the threshold that we set (here we chose 0.075, but feel free to experiment) The value we chose masks most of the ocean and only a little bit of the land.

```
gmt grdmath corr_stack.grd 0.075 GE 0 NAN = mask_def.grd
```

Use ncview again to view the mask_def.grd file to inspect the mask. If it masks too much (lots of white patches on land, see white figure), then lower the coherence threshold. If it masks too little (for example, if there is speckle in the ocean, see right figure), raise it. The middle figure here shows a 0.075 threshold:



iii. Link the new mask_def.grd to all /date1-date2/ directories and **unwrap each interferogram**. This is the same as unwrapping using the script above (part a), but this time with the link to mask_def.grd added in. Remember to make a list (intflist) of interferogram directories to feed the script. Make the following script and name it "**unwrap_intf.csh**":

```
________________________________________
#!/bin/csh -f
#  intflist contains a list of all date1_date2 directories.

foreach line (`awk '{print $1}' intflist`)
  cd $line
  ln -s ../mask_def.grd .
  snaphu_interp.csh 0.001
  cd ..
end
________________________________________
```

See the end of part (a) for information on running the unwrapping in parallel or adding a region cut to your script.

11. Run **SBAS**

a) In **SBAS** directory, prepare **scene.tab** fie. The first column is the scene ID and the second column is the number of days

```
awk '{print $1 , $3}' ../F1/baseline_table.dat >scene.tab
vi scene.tab
```

:%s/_F1/ /g

Example of what scene.tab should look like:

<scene ID>	<Days>
S1_20161013_ALL	1016
S1_20161106_ALL	1040
S1_20161118_ALL	1052
S1_20161130_ALL	1064
S1_20161224_ALL	1088
S1_20170117_ALL	1111
S1_20170210_ALL	1135

b) In **SBAS** directory, Prepare **intf.tab**.

Example of intf.tab:

<full path to unwrap.grd>	<full path to corr.grd>	<ID date1>	<ID date2>	<b_perp>
../merge/2016286_2016310/unwrap.grd	../merge/2016286_2016310/corr.grd	S1_20161013_ALL	S1_20161106_ALL	21.39795
../merge/2016286_2016322/unwrap.grd	../merge/2016286_2016322/corr.grd	S1_20161013_ALL	S1_20161118_ALL	25.47900
../merge/2016286_2016328/unwrap.grd	../merge/2016286_2016328/corr.grd	S1_20161013_ALL	S1_20161124_ALL	-41.1191
../merge/2016310_2016322/unwrap.grd	../merge/2016310_2016322/corr.grd	S1_20161106_ALL	S1_20161118_ALL	75.08018
../merge/2016310_2016328/unwrap.grd	../merge/2016310_2016328/corr.grd	S1_20161106_ALL	S1_20161124_ALL	-62.51705
../merge/2016310_2016340/unwrap.grd	../merge/2016310_2016340/corr.grd	S1_20161106_ALL	S1_20161206_ALL	42.91977

The b_perp value is the perpendicular baseline of the first date minus the perpendicular baseline of the second date. This is found in baseline_table.dat.

c) Run SBAS

i) Type "sbas" to see list of arguments and how to structure the command.

Here is a sample command for Kilauea:

```
sbas intf.tab scene.tab 38 10 2313 2259 -range 888126 -incidence 40 -wavelength 0.0554658 -smooth 5.0 -rms -dem
```

- To find xdim and ydim (x and y dimension of interferograms):
  gmt grdinfo ../merge/date1_date2/unwrap.grd
  "x n_columns" is the value for x dim
  "y n_rows" is the value for y dim
- wavelength: wavelength of the radar wave in m. To find this out, look in supermaster.PRM for "radar_wavelength"
- range: range distance from radar to center of interferogram. To get this number for Sentinel-1:

  ((speed of light) / (rng_samp_rate) / (x_min+x_max)) + 845000

  - rng_samp_rate is in supermaster.PRM

sample calculation:
3e8/64345238.125714/36995.5895372+845000 = 888,126.734485

ii) After running SBAS, you get a vel.grd file (in radar coordinates). To transform it into lat/long, do the following:

```
ln -s ../topo/trans.dat .
ln -s ../intf_all/2017134_2017158/gauss_*
proj_ra2ll.csh trans.dat vel.grd vel_ll.grd
gmt grd2cpt vel_ll.grd -T= -Z -Cjet > vel_ll.cpt
grd2kml.csh vel_ll vel_ll.cpt
```