**This requires ISCE2 with stack processor. I always follow this guide for installation of isce, mintpy** [**https://github.com/yunjunz/conda-envs**](https://github.com/yunjunz/conda-envs)

**DEM**

$ sardem --bbox -107 38 -104 40 --data COP -isce

% NOTE for this UAVSAR processing, we wont actually use the DEM here, but you need it to run prepareUAVSAR\_coregStack.py. We will use the DEM provided with the UAVSAR data.

**stripmapStack**

1. Download SLC Stack files (.slc, .ann, and .dop) from <https://uavsar.jpl.nasa.gov>.

Graphical user interface

Description automatically generated

1. copy the wget commands from the UAVSAR into a terminal

Graphical user interface, text, application

Description automatically generated

1. Put everything except for the .dop file inside a directory named “download”
2. **run**:

$ prepareUAVSAR\_coregStack.py -i download -o SLC -d swatch\_00540\_01\_BC.dop -s 1

i = directory where data are downloaded, o = SLC folder, d = .dop file. Note that if there are multiple segments need to add command “- s 2” or “-s 3” . Segment 1 is default.

1. create a “merged” directory and then create a soft link inside it to the SLC directory (inside the “merged” directory run: ln -s ../SLC )
2. **run:**

$ stackStripMap.py -s SLC -d /u/sar-r1/handwerg/insar/Colorado/AspenArea/DEM/DEM\_GLO30/elevation.dem -a 8 -r 3 -W interferogram --nofocus -S uavsar\_stack -t 1100 -u snaphu --filter\_strength 0.4

* 1. set azimuth and range looks, and make sure that the time in days is enough large to cover all the time for a dense stack
  2. This step creates the run files you need. You only will run a single run file to create the interferograms

1. **create interferograms:**

$ source run\_08\_igram

(make sure to run it inside a screen, could take a while)

1. Use jupyter notebooks to create the geometry files. Start with 1\_prep\_uavsar\_geometry. This notebook will take the geometry files from UAVSAR, then re-interpolate them to 1x1 looks so that you can then make them for the specific number of looks used in your processing.
2. Use juypter notebook 2\_prep\_isce\_MintPy\_UAVSAR.ipynb to create mintpy files for UAVSAR(if this is preferred). This is then used to geocode.
3. Geocode the data using notebook 3\_geocode.ipynb

**Currently there is only a single pair of interferograms so no need for time series. Simply run the data into mintpy for formatting purposes and geocoding.**

**Calculate spatial coherence.**

temporal\_average.py ifgramStack.h5 -d coherence -o avgSpatialCoh.h5

**Pixel tracking**

python denseoffset\_adj.py -m ../SLC/20230725/20230725.slc -s ../SLC/20230925/20230925.slc

window size and steps are hard coded into the denseOffset script

geocode using 4\_geocode\_offsets.ipynb

\*\* you will need to make another set of the geometry files using the run\_mulitlook\_geometry.sh where the looks = skips.