Using stackSentinel.py from ISCE2

**Installation:**

1. Check this instruction from Yunjun: <https://github.com/earthdef/sar-proc>
2. Install ISCE2. Follow <https://github.com/yunjunz/conda_envs>
   1. Install from conda\_forge: **opt1** of the workflow
   2. Install from source (developer mode): **opt2** of the workflow

**Downloading files:**

1. Download L1 Single look complex (SLC) from ASF vertex (your ROI):
   1. User interface from <https://search.asf.alaska.edu/#/>
   2. [ssara\_federated\_query.py](https://www.unavco.org/gitlab/unavco_public/ssara_client) with command line parallel downloading, ex.  
      ssara\_federated\_query.py --platform=SENTINEL-1A -i 'YOUR\_POLYGON\_HERE' -r 94 --flightDirection D --download --parallel 8
2. [Download antenna pattern auxiliary files](https://github.com/isce-framework/isce2/tree/main/contrib/stack/topsStack#aux_cal-file-download) (before 2015)

**Processing stack of SLCs:**

**(**[**run\_isce\_stack.py**](https://github.com/earthdef/sar-proc#2-stack-processing-on-kamb-via-run_isce_stackpy)will automatically takes care of the following):

1. [Download DEM](https://github.com/isce-framework/isce2/tree/main/contrib/stack/topsStack#2-prepare-dem) from SRTM (specifying a larger bounding box): specify 1-arcsec resolution
2. Call [stackSentinel.py](https://github.com/isce-framework/isce2/tree/main/contrib/stack/topsStack#sentinel-1-tops-stack-processor) and generate the run files. Specify:  
   -s $SLC\_DIR -d $DEM -o $ORBIT\_DIR -a $AUX\_DIR -b $BBOX -c 3   
   --filter\_strength 0 --azimuth\_looks $ALOOKS (20) --range\_looks $RLOOKS (5)  
   --numprocess4topo 8 --referencedate 20200101 --param\_ion ./ion\_param.txt  
   --num\_connections\_ion 3 –useGPU
3. Go inside `runfiles` folder. Run all the bash run files sequentially.
4. Finally, we processed the SLC. Now we can start feed the interferograms into `MintPy` for time-series analysis.

**Other notes:**

* Choose your CUDA device and number of threads:
  + export CUDA\_VISIBLE\_DEVICES=1
  + export OMP\_NUM\_THREADS=4
* Want to process with Campus HPC
  + Get the code and files from: <https://github.com/earthdef/sar-proc/tree/main/hpc_isce_stack>
  + Same steps from 1~5. Replace step 6 & 7 with the following:
  + Modify the following based on your need:
    - [stack\_sentinel\_cmd.sh](https://github.com/earthdef/sar-proc/blob/main/hpc_isce_stack/stack_sentinel_cmd.sh)
    - [resources\_array.cfg](https://github.com/earthdef/sar-proc/blob/main/hpc_isce_stack/resources_array.cfg)
    - [write\_sbatch\_files\_array.py](https://github.com/earthdef/sar-proc/blob/main/hpc_isce_stack/write_sbatch_files_array.py) (email address)
  + Run [stack\_sentinel\_cmd.sh](https://github.com/earthdef/sar-proc/blob/main/hpc_isce_stack/stack_sentinel_cmd.sh) to generate the run files and sbatch files
  + Go inside `runfiles` folder. Run [submit\_chained\_dependencies.sh](https://github.com/earthdef/sar-proc/blob/main/hpc_isce_stack/submit_chained_dependencies.sh)
  + Use `squeue -u usaer\_name` to track status