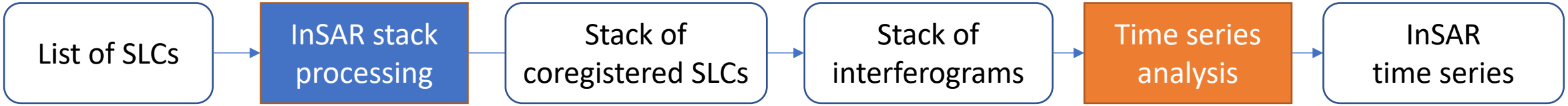
Stack processing with ISCE2

Heresh Fattahi, Yunjun Zhang

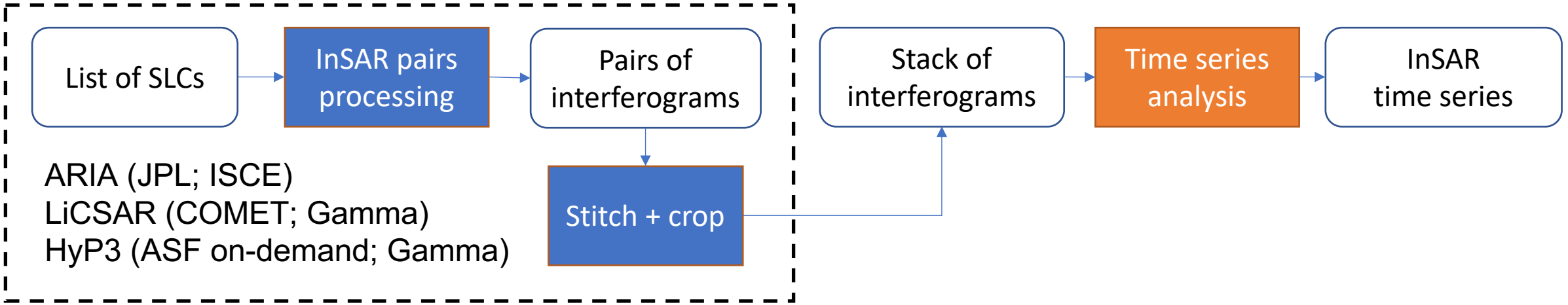
Aug 27, 2021

Stack processing and pairs processing

- isce2: topsApp, stripmapApp, alos2App



- isce2: topsStack, stripmapStack, alosStack



Stack processing

Why stack processing:

In order to do time-series analysis a stack of pairs of interferograms/offsets/corrections are needed

Two ways to generate the stacks:

Option-1 (pairwise) : run pair processing applications many times, generate pairs that are not aligned, have costume scripts to align them or make sure they are geocoded to the same grid

Option 2 (stack): align the SLCs at the beginning, so interferograms naturally are aligned

Option 1 needs to run topo many times (number of pairs)

Note: topo (i.e., radar to geo transformation) is the most time consuming step.

Option 2 runs topo only one time! Significantly more efficient!

Option 2 ensures consistency withing the stack and reduces the risk of extra inconsistency (processing errors). It's a more precise alignment.

Option 2 also allows to generate coregistered SLC stack for PS analysis methods.

Stack processors in isce2

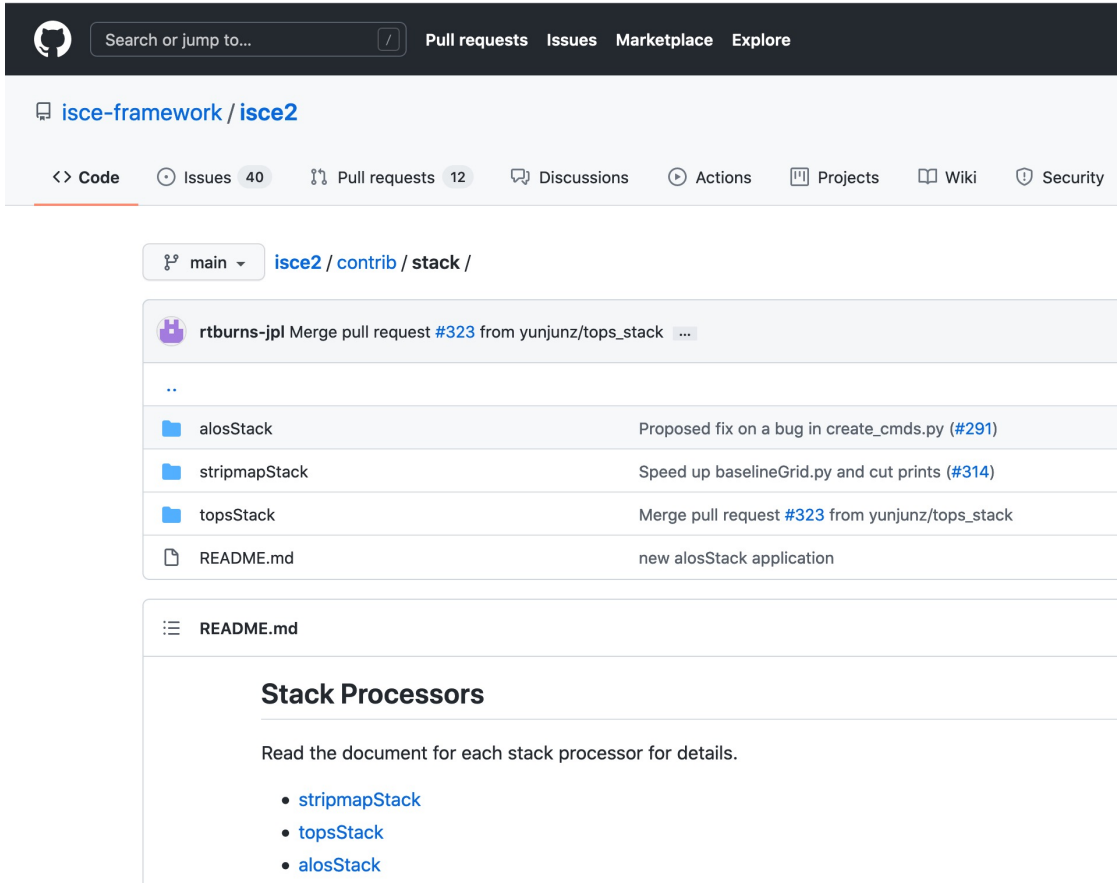
WARNING:
Stack processors should be used by advanced users with enough knowledge about interferometry, InSAR processing steps and ISCE2 pair processing, working in a linux environment and Python programming.

ISCE2 includes three stack processors for processing stacks of:

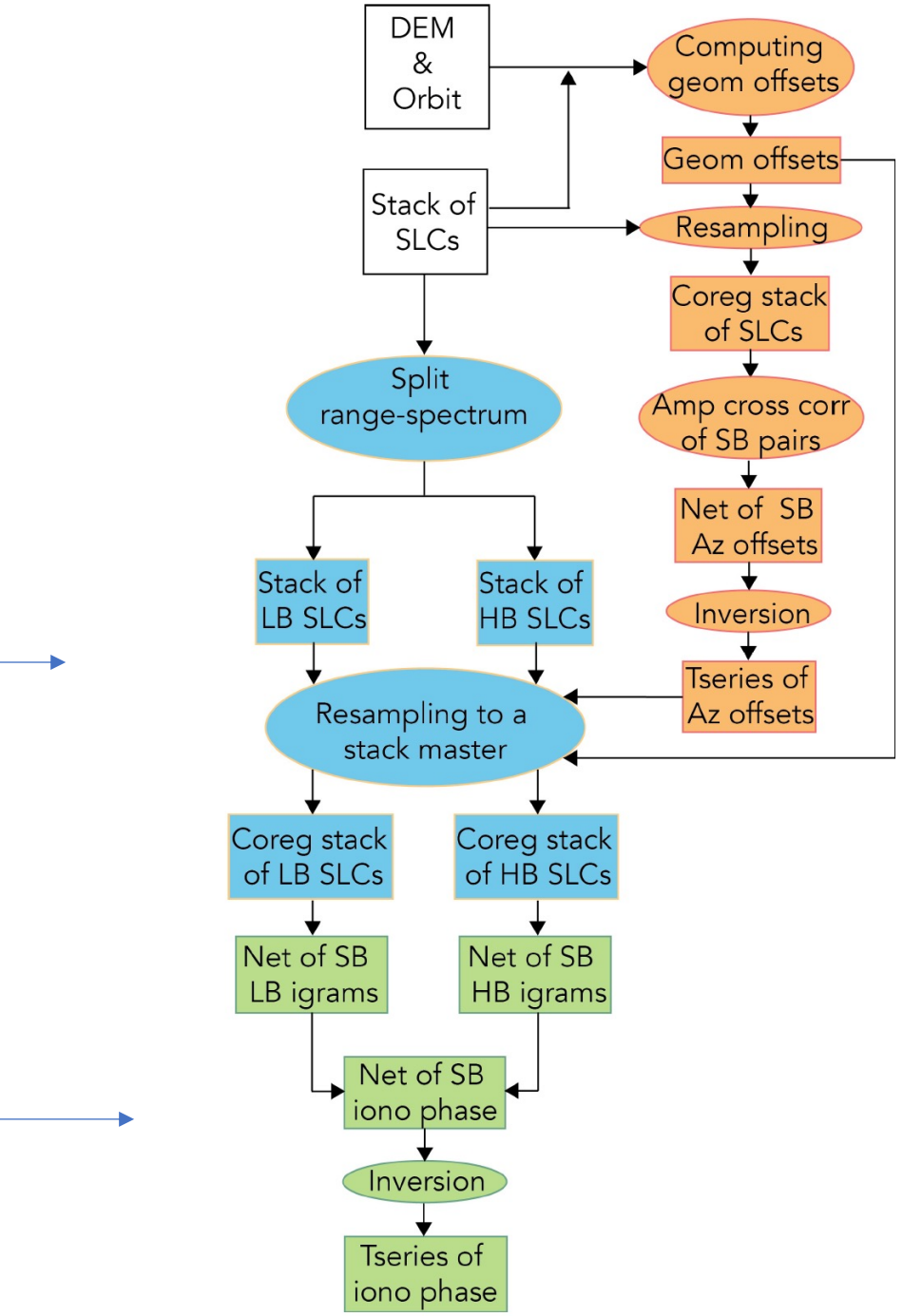
- Stripmap
- TOPS
- ScanSAR

Stack processors may produce stacks of coregistered SLCs or interferograms

For an interferogram stack, first the stack of SLCs get coregistered to a stack reference image.



Stripmap stack processor



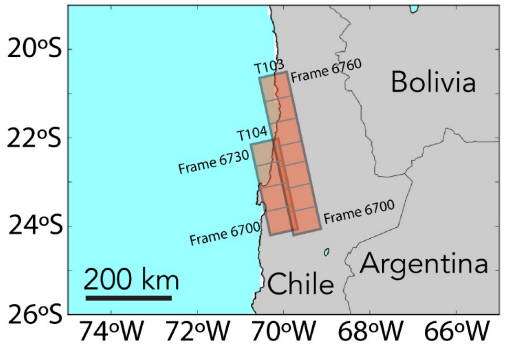
Coregistration block

Split spectrum block

Interferogram formation and Ionospheric phase estimation

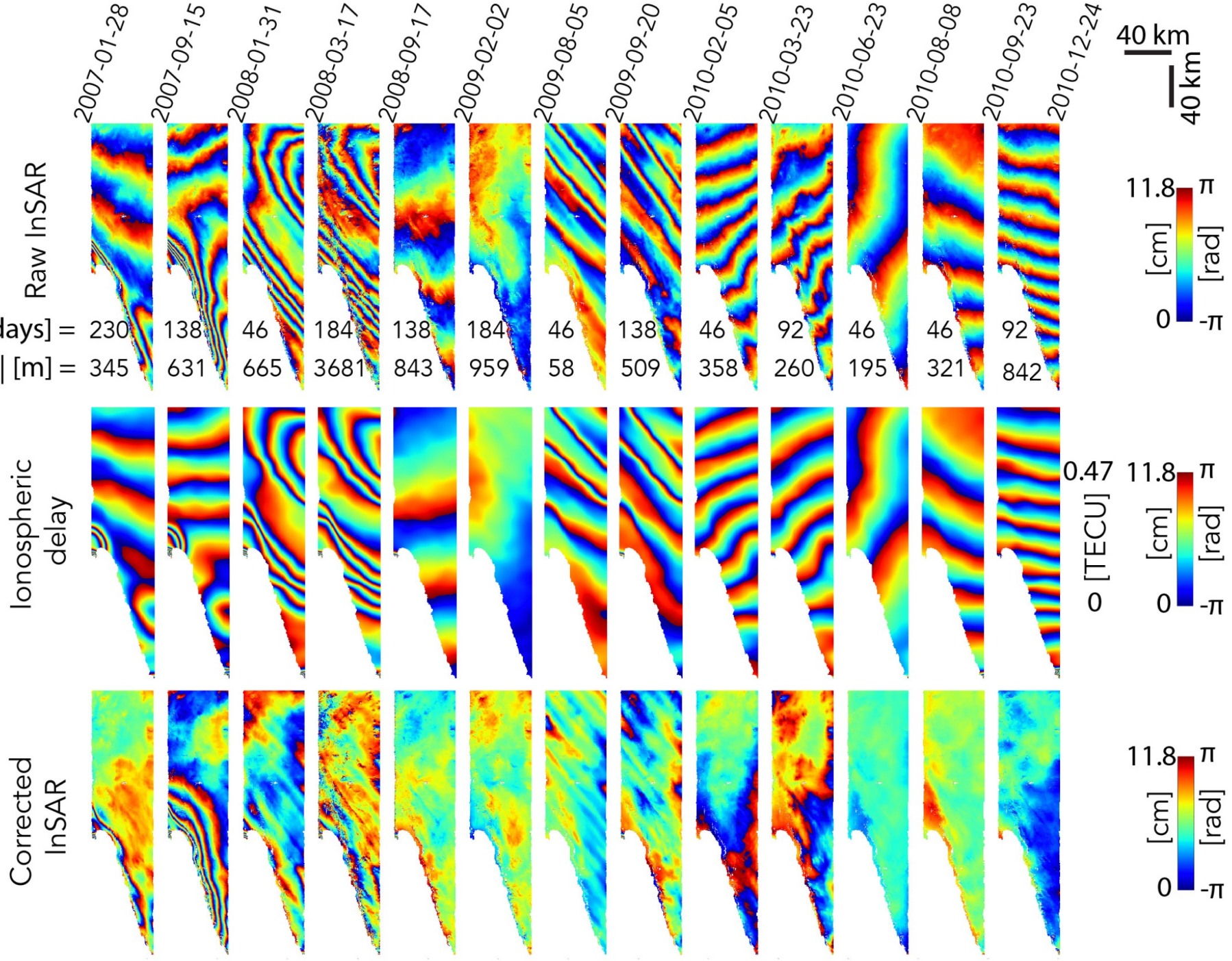
[*Fattahi et al, 2017*]

Example stack of ALOS-1



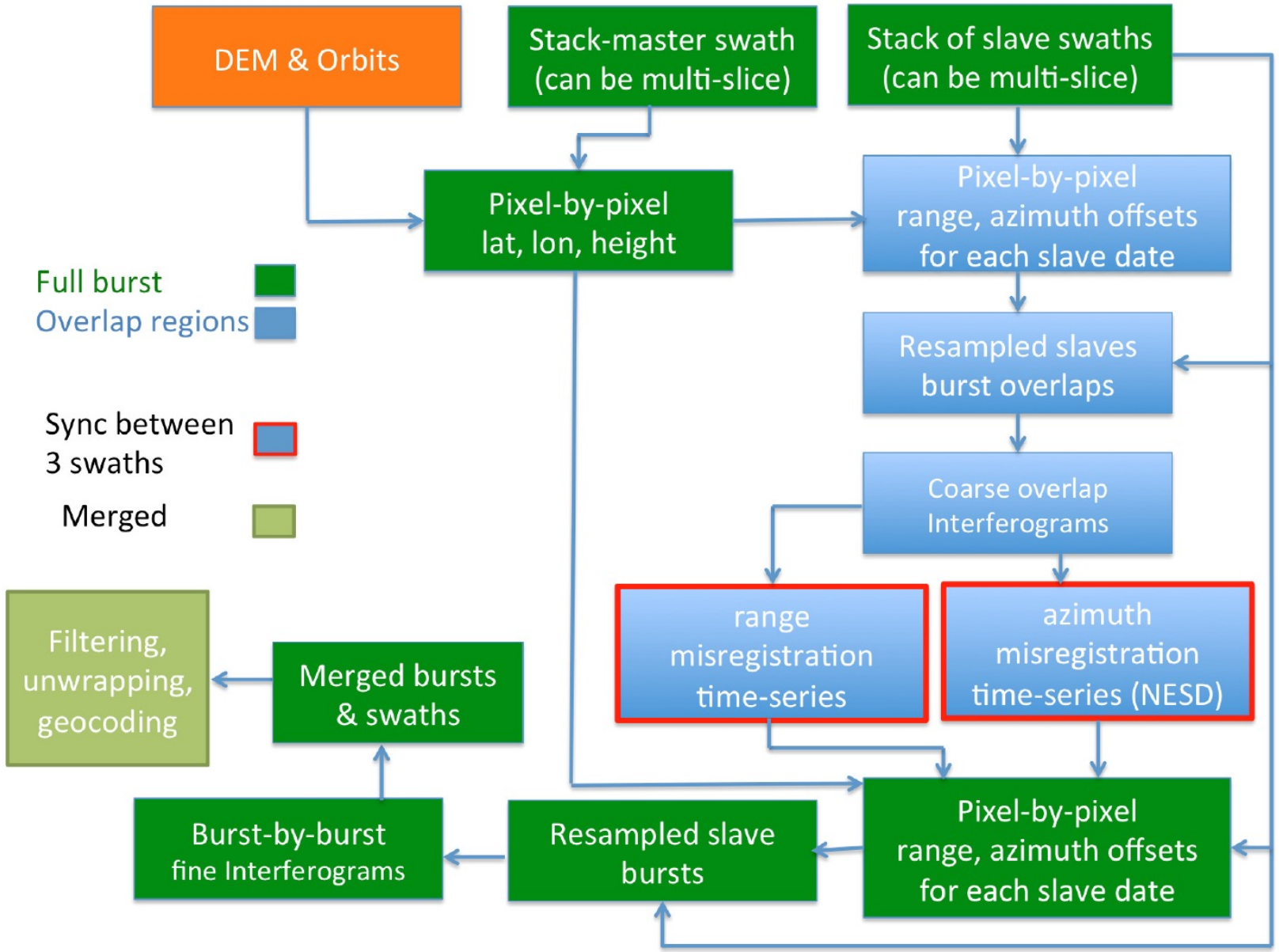
A stack of interferograms and ionospheric phases were inverted to form tim-series

[*Fattahi et al, 2017*]



TOPS stack processor

The stack processor allows to choose between the different coregistration methods (geometry or ESD based)



[*Fattahi et al, 2017*]

Resources

Stack processors on public github:

https://github.com/isce-framework/isce2/tree/main/contrib/stack

Step by step instruction in README (stripmap stack):

https://github.com/isce-framework/isce2/tree/main/contrib/stack/stripmapStack

Step by step instruction in README (TOPS stack):

https://github.com/isce-framework/isce2/tree/main/contrib/stack/topsStack

Step by step instruction in README (scanSAR stack):

https://github.com/isce-framework/isce2/blob/main/contrib/stack/alosStack/alosStack_tutorial.txt

Get access to TOPS Stack Processor on OpenSARLab:

1. Open a terminal on OpenSARLab

2. Activate the “unavco” environment:
   conda activate unavco

3. Add the path of the TOPS stack processor to your PATH variable:

   export PATH=$ISCE_STACK/topsStack:${PATH}

Check your steup by running:
stackSentinel.py --help

NOTE:
Same approach works when you have your own installation of isce2 on your local machine or cluster

Get access to Stripmap Stack Processor on OpenSARLab:

1. Open a terminal on OpenSARLab

2. Activate the “unavco” environment:
   conda activate unavco

3. Add the path of the stack processor to your PATH variable:

   export PATH=$ISCE_STACK/stripmapStack/:${PATH}

Check your steup by running:
   stackStripMap.py --help

NOTE:
Same approach works when you have your own installation of isce2 on your local machine or cluster. In this case you may replace “unavco” with any environment you have used to install isce2 with conda.

Get access to ScanSAR Stack Processor on OpenSARLab:

1. Open a terminal on OpenSARLab

2. Activate the “unavco” environment:
   conda activate unavco

3. Add the path of the stack processor to your PATH variable:

   export PATH=$ISCE_STACK/alosStack/:${PATH}

Check your steup by running:
create_cmds.py --help

NOTE:
Same approach works when you have your own installation of isce2 on your local machine or cluster. In this case you may replace “unavco” with any environment you have used to install isce2 with conda.

References

H. Fattahi, M. Simons, and P. Agram, "InSAR Time-Series Estimation of the Ionospheric Phase Delay: An Extension of the Split Range-Spectrum Technique", IEEE Trans. Geosci. Remote Sens., vol. 55, no. 10, 5984-5996, 2017. (https://ieeexplore.ieee.org/abstract/document/7987747/

H. Fattahi, P. Agram, and M. Simons, “A network-based enhanced spectral diversity approach for TOPS time-series analysis,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 2, pp. 777–786, Feb. 2017. (https://ieeexplore.ieee.org/abstract/document/7637021/)

Liang and E. J. Fielding, "Interferometry with ALOS-2 full-aperture ScanSAR data,"
IEEE Transactions on Geoscience and Remote Sensing, vol. 55, no. 5, pp. 2739-2750, May 2017.