



## SENTINEL-1

# Command-Line InSAR Processing

Issued February 2020

Alex McVittie

# Introduction

This tutorial will guide you through the setup of creating an InSAR processing framework for Sentinel-1 SLC radar imagery to create interferograms.

## System requirements

Interferogram generation is resource-intensive. Having at minimum a quad-core CPU, 16gb of RAM and 20gb of free disk space is recommended. You also need to have an EarthData (Free to register) account to download the Sentinel-1 imagery. As the scenario in this tutorial is from 2016, it is in offline mode on the Copernicus data hub, increasing wait time to access. NASA's ASF Vertex has the data readily available.

You will also need to have SNAP installed on a desktop with a monitor to set up the processing graph, prior to running this on a server with command-line only.

## Scenario

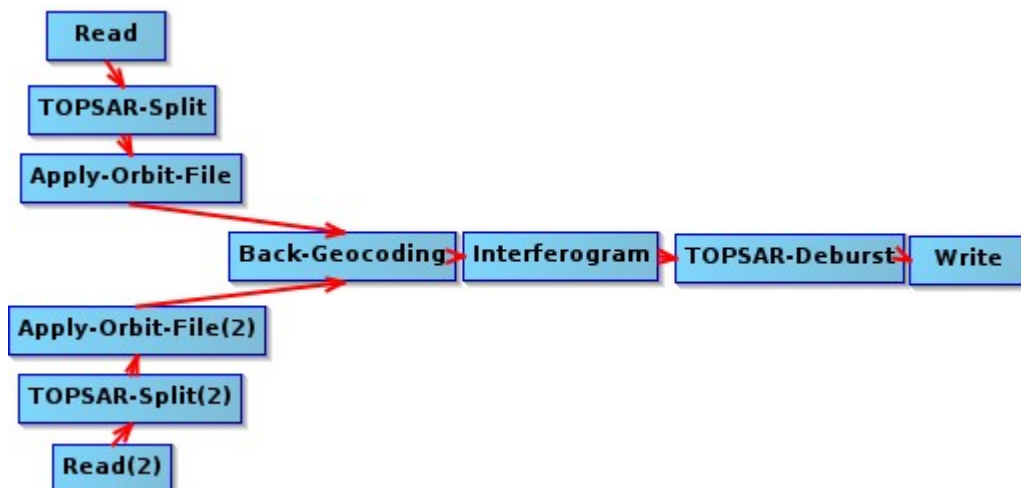
For this InSAR tutorial, we will be generating an interferogram that visualizes the surface deformation caused by the April 16<sup>th</sup> 2016 Kumamoto earthquake in Japan. Two Sentinel-1 products are needed, one for before the event, and one for after. They can be downloaded here:

[https://datapool.asf.alaska.edu/SLC/SA/S1A\\_IW\\_SLC\\_1SSV\\_20160408T091355\\_20160408T091430\\_010728\\_01001F\\_83EB.zip](https://datapool.asf.alaska.edu/SLC/SA/S1A_IW_SLC_1SSV_20160408T091355_20160408T091430_010728_01001F_83EB.zip)

[https://datapool.asf.alaska.edu/SLC/SA/S1A\\_IW\\_SLC\\_1SSV\\_20160420T091355\\_20160420T091423\\_010903\\_010569\\_F9CE.zip](https://datapool.asf.alaska.edu/SLC/SA/S1A_IW_SLC_1SSV_20160420T091355_20160420T091423_010903_010569_F9CE.zip)

## Building the processing graph

We will be building upon an existing graph that is provided with SNAP – the TOPSAR Coreg Interferogram graph, available in the graph builder through *Graphs > InSAR Graphs*.

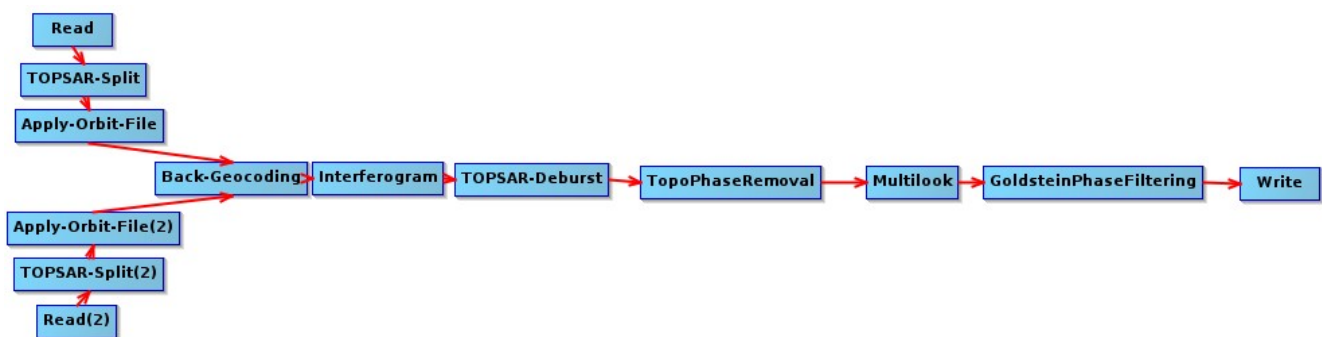


While this generates an initial interferogram and debursts it, there are a few enhancement steps required after the TOPSAR-Deburst. Topographic phase removal must be run, along with multi-looking and Goldstein phase filtering. Topographic phase removal (*Radar > Interferometric > Products > TopoPhaseRemoval*) improves the accuracy of the interferogram by reducing the influence created by terrain using a DEM. SNAP will auto-select a DEM to download, but you may wish to modify the parameters to choose a different DEM in the processor options.

At this point, the interferogram has a large amount of noise, with patterns in the image being difficult to detect. Multi-looking (*Radar > SAR Utilities > Multilook*) will improve the phase fidelity of the the interferogram, which performs averaging of neighbouring pixels.

Within the processing parameters tab, the number of range and azimuth looks defines the window size that averages the product. Setting the Range Looks to 6 and the Azimuth Looks to 2 results in a 6x2 window to average pixels with, creating an approximately 25x25m resolution raster.

Finally, we must apply the Goldstein phase filter (*Radar > Interferometric > Filtering > GoldsteinPhaseFiltering*) to improve the interferometric phase. Our graph should now look like the following:



Save this graph XML file and open the XML file in your preferred text editor. To build a repeatable workflow, we want to substitute a few values in the graph with variables that can be brought in from a text file – this allows you to run this graph in batch on many sentinel-1 image pairs, if you needed to produce many sets of interferograms. SNAP’s gpt can substitute `#{variableName}` with a variable value provided in a configuration file.

Within the first Read, add the following directly after `<parameters/>` :

```

<parameters class="com.bc.ceres.binding.dom.XppDomElement">
<file>#{product1}</file></parameters>
  
```

If this already exists due to having products open in SNAP and the reads being auto-populated, replace the path in the `<file>` tags with the `#{product1}` variable.

Do the same thing with the second Read, but have `#{product2}` in the `<file>` tags instead.

Replace `<subswath/>` in both TOPSAR-Split operators with `<subswath>#{subswath}</subswath>`, along with replacing `<selectedPolarisations/>` with `<selectedPolarisations>#{polarization}</selectedPolarisations>`

Additionally, within the TOPSAR-Split operators, remove the `firstBurstIndex` and `lastBurstIndex` tags and values.

Within Multilook, replace `<nRgLooks>6</nRgLooks>` with `<nRgLooks>${rangeLook}</nRgLooks>` and `<nAzLooks>2</nAzLooks>` with `<nAzLooks>${azLook}</nAzLooks>`

Within Write, replace `<file>/tmp/target.dim</file>` with `<file>${outputFile}</file>`

With our graph XML file prepared, let's create a parameter file that we can use to give SNAP the information it needs to transform our raw products into a filtered interferogram.

Create a new text document titled `input_parameters.properties`. Within it, populate it as such, replacing the paths for the two products with the path to where you have saved your products:

```
product1=/media/skywatch/USB/2016/
S1A_IW_SLC__1SSV_20160408T091355_20160408T091430_010728_01001F_83EB.zip

product2=/media/skywatch/USB/2016/
S1A_IW_SLC__1SSV_20160420T091355_20160420T091423_010903_010569_F9CE.zip

subswath=IW1

selectedPolarisations=VV

rangeLook=6

azLook=2

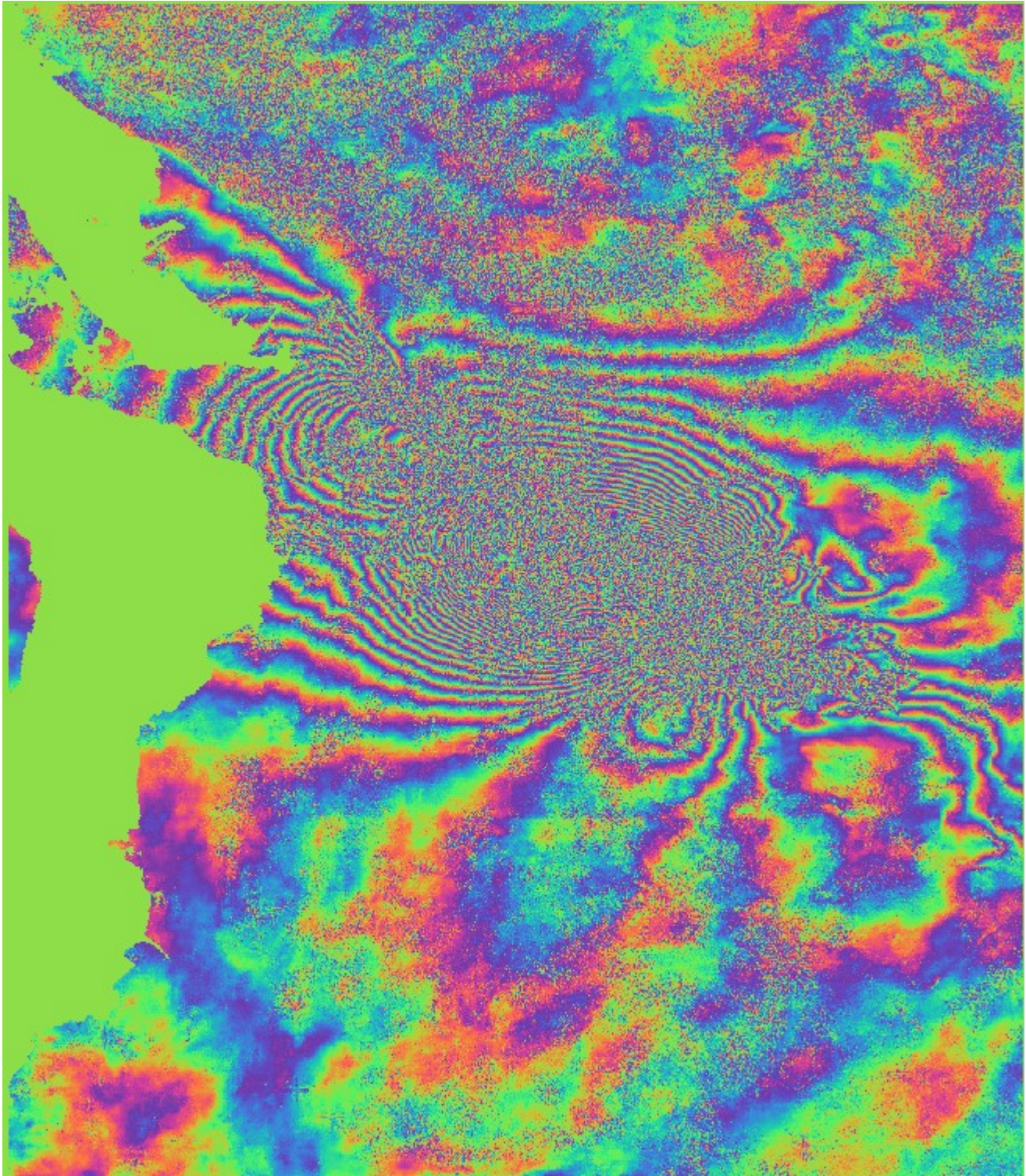
outputFile=/data/japan_earthquake.dim
```

To execute this, in a shell simply run

```
gpt processing_graph.xml -e -p input_parameters.conf
```

and it will write the filtered interferogram out to (in this case) `/data/japan_earthquake.dim`

The GPT processing framework has many additional options that you can set. Run `gpt -h` to see a full list of options, such as setting available RAM and multithreading options.



*Output interferogram – the deformation fringe pattern very clearly shows the areas affected by the earthquake.*