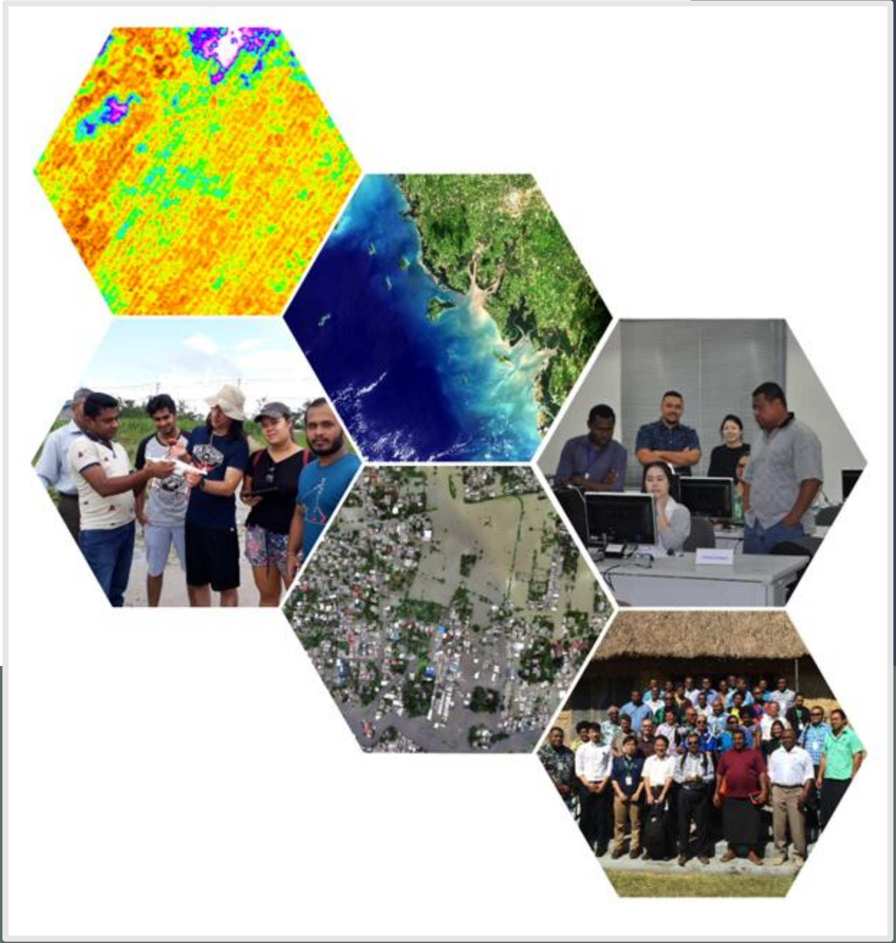


# Urban Flood Mapping using Sentinel-1: Coherence Changes

syams@ait.asia

Geoinformatics Center - AIT

# Overview

## Flood in Central Java, Indonesia on March 2024

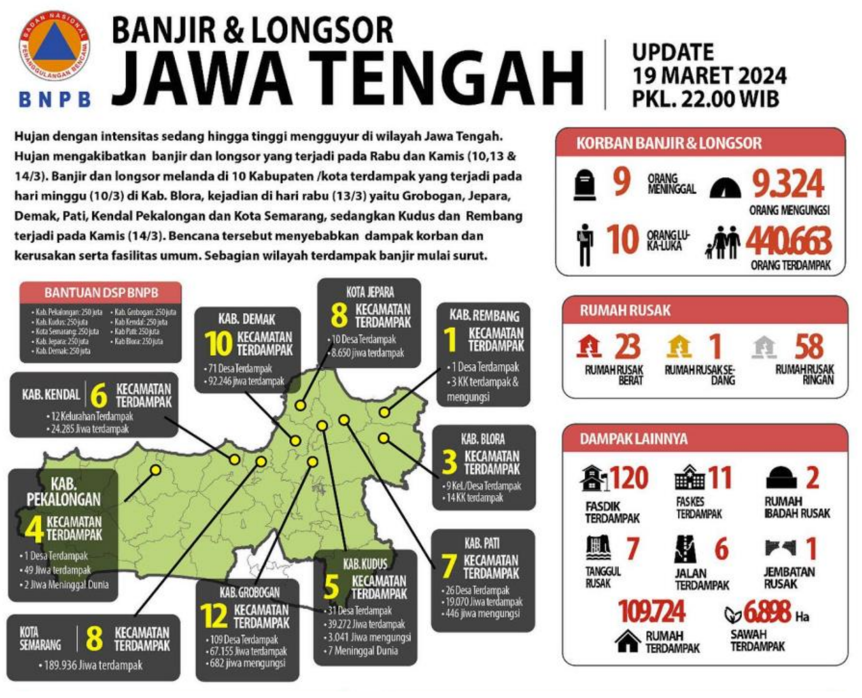
- Extreme weather and heavy rainfall caused floods and flash floods in Central Java, which overwhelmed river systems and hydrological infrastructure (dam failures) across the northern coast region.
- In some areas, heavy rainfall has started since February and continues until mid to the end of March.
- Affected areas include Kota Semarang, Kota Jepara, Pekalongan, Kendal, Demak, Rembang, Blora, Pati, Kudus, and Grobogan (*BPBD Jateng)*.
- Damages to public infrastructure and hundreds of thousands of residents were affected by the floods.

*Kecamatan Karanganyar, Kebupaten Demak, Jawa Tengah, Sunday (17/3/2024)*

- Sentinel Asia was activated on 20 March 2024 at the request of the National Research and Innovation Agency (BRIN).
- Archive and crisis images were obtained from multiple sources, including ALOS-2 from the Japan Aerospace Exploration Agency (JAXA), FORMOSAT-5 from the Taiwan Space Agency (TASA), TelEOS-1 from the Centre for Remote Imaging Sensing and Processing (CRISP), and Resourcesat-2 from the Indian Space Research Organization (ISRO).

# Overview

## Flood in Central Java, Indonesia on March 2024



BNPB

**BANJIR & LONGSOR JAWA TENGAH**

UPDATE 19 MARET 2024 PKL. 22.00 WIB

**Hujan dengan intensitas sedang hingga tinggi mengguyur di wilayah Jawa Tengah. Hujan mengakibatkan banjir dan longsor yang terjadi pada Rabu dan Kamis (10,13 & 14/3). Banjir dan longsor melanda di 10 Kabupaten /kota terdampak yang terjadi pada hari minggu (10/3) di Kab. Blora, kejadian di hari rabu (13/3) yaitu Grobogan, Jepara, Demak, Pati, Kendal Pekalongan dan Kota Semarang, sedangkan Kudus dan Rembang terjadi pada Kamis (14/3). Bencana tersebut menyebabkan dampak korban dan kerusakan serta fasilitas umum. Sebagian wilayah terdampak banjir mulai surut.**

**BANTUAN DSP BNPB**
- Kab. Pekalongan: 250 juta
- Kab. Kudus: 250 juta
- Kota Semarang: 250 juta
- Kab. Jepara: 250 juta
- Kab. Demak: 250 juta
- Kab. Grobogan: 250 juta
- Kab Kendal: 250 juta
- Kab Pati: 250 juta
- Kab Blora: 250 juta

**KAB. DEMAK** 10 KECAMATAN TERDAMPAK
- 71 Desa Terdampak
- 92.246 jiwa terdampak

**KOTA JEPARA** 8 KECAMATAN TERDAMPAK
- 10 Desa Terdampak
- 8.650 jiwa terdampak

**KAB. REMBANG** 1 KECAMATAN TERDAMPAK
- 1 Desa Terdampak
- 3 KK terdampak & mengungsi

**KAB. KENDAL** 6 KECAMATAN TERDAMPAK
- 12 Kelurahan Terdampak
- 24.285 Jiwa terdampak

**KAB. BLORA** 3 KECAMATAN TERDAMPAK
- 9 Kel./Desa Terdampak
- 14 KK terdampak

**KAB. PEKALONGAN** 4 KECAMATAN TERDAMPAK
- 1 Desa Terdampak
- 49 Jiwa terdampak
- 2 Jiwa Meninggal Dunia

**KAB. PATI** 7 KECAMATAN TERDAMPAK
- 26 Desa Terdampak
- 19.070 Jiwa terdampak
- 446 jiwa mengungsi

**KAB. KUDUS** 5 KECAMATAN TERDAMPAK
- 31 Desa Terdampak
- 39.272 Jiwa terdampak
- 3.041 Jiwa mengungsi
- 7 Meninggal Dunia

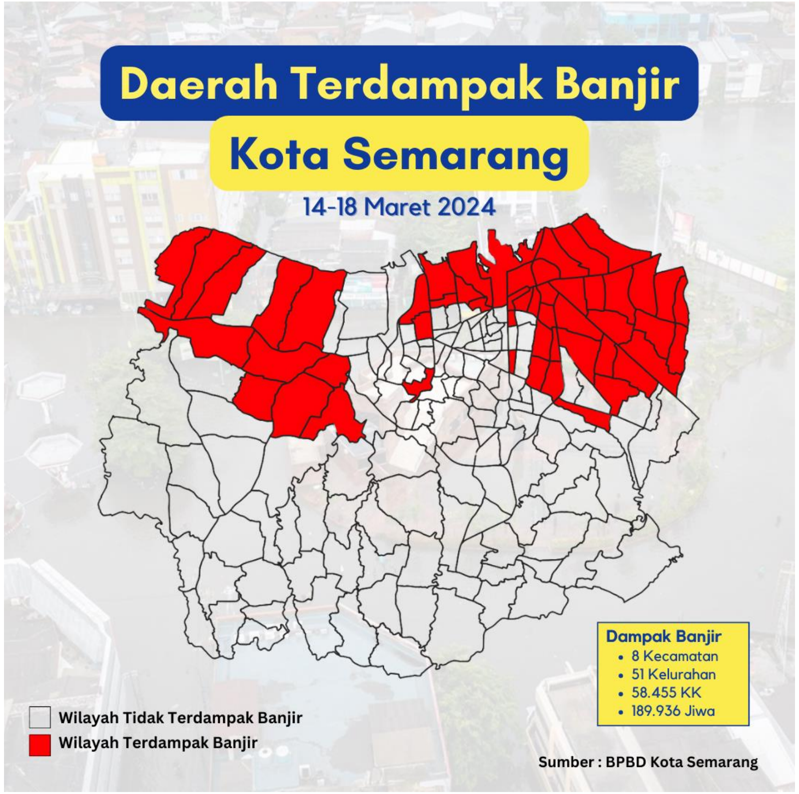
**KAB. GROBOGAN** 12 KECAMATAN TERDAMPAK
- 109 Desa Terdampak
- 67.155 Jiwa terdampak
- 682 jiwa mengungsi

**KOTA SEMARANG** 8 KECAMATAN TERDAMPAK
- 189.936 Jiwa terdampak

**KORBAN BANJIR & LONGSOR**
- 9 ORANG MENINGGAL
- 9.324 ORANG MENGUNGSI
- 10 ORANG LUKA-LUKA
- 440.663 ORANG TERDAMPAK

**RUMAH RUSAK**
- 23 RUMAH RUSAK BERAT
- 1 RUMAH RUSAK SEDANG
- 58 RUMAH RUSAK RINGAN

**DAMPAK LAINNYA**
- 120 FASDIK TERDAMPAK
- 11 FASKES TERDAMPAK
- 2 RUMAH IBADAH RUSAK
- 7 TANGGUL RUSAK
- 6 JALAN TERDAMPAK
- 1 JEMBATAN RUSAK
- 109.724 RUMAH TERDAMPAK
- 6.898 Ha SAWAH TERDAMPAK



## Overview

### Objective

The objective of this exercise is to generate an urban flood map by analyzing coherence changes derived from pre- and post-event Sentinel-1 SLC (Single Look Complex) data.

This exercise aims to detect areas affected by flooding based on significant coherence loss, supporting rapid disaster response and impact assessment in urban environments.

***Note**: The exercise is adapted from the UN-SPIDER Recommended Practice for Flood Mapping with Sentinel-1 Interferometric Coherence*

## Overview

### Data and Software

In this exercise, we will use SNAP software and Google Earth Engine (GEE).

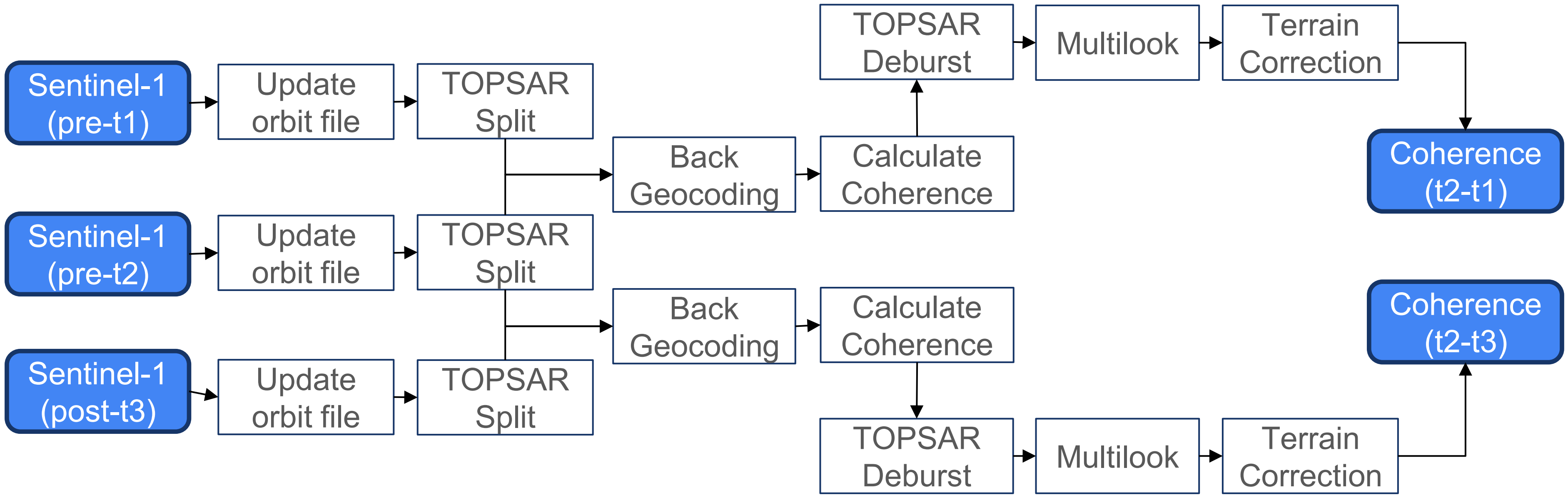
The following data are available:

- Sentinel-1 SLC (Single Look Complex)
  - Contain magnitude and phase information represented by complex I and Q numbers. Range coordinate is in slant range.
  - Sentinel-1 Terrain Observation by Progressive Scans SAR (TOPS) captures data in bursts, which are segments of radar echoes acquired by cyclically switching the antenna beam across multiple sub-swaths. Sentinel-1 SLC data are split into bursts per sub-swath.
  - Observation mode: Interferometric Wide (IW).
  - Dual Polarization (VH and VV), Data format: SAFE structure
  - Date: 15 March 2024 (observe), 10 and 22 December 2023 (archive)

# Methodology

## Urban flood mapping using Sentinel-1 coherence changes

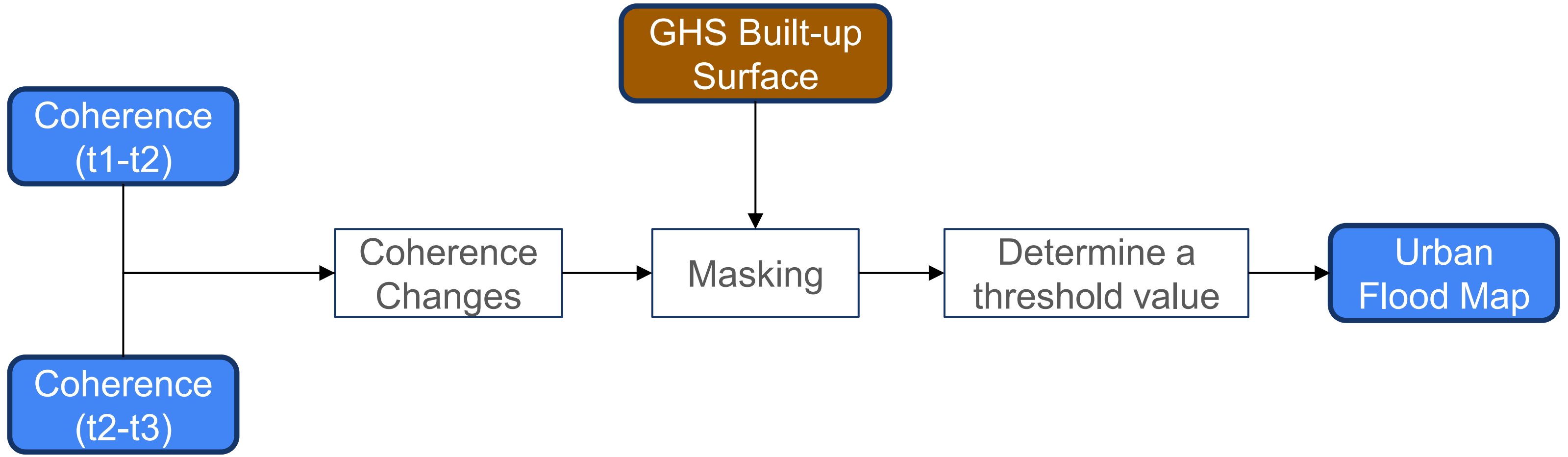
**Processing using SNAP**



# Methodology

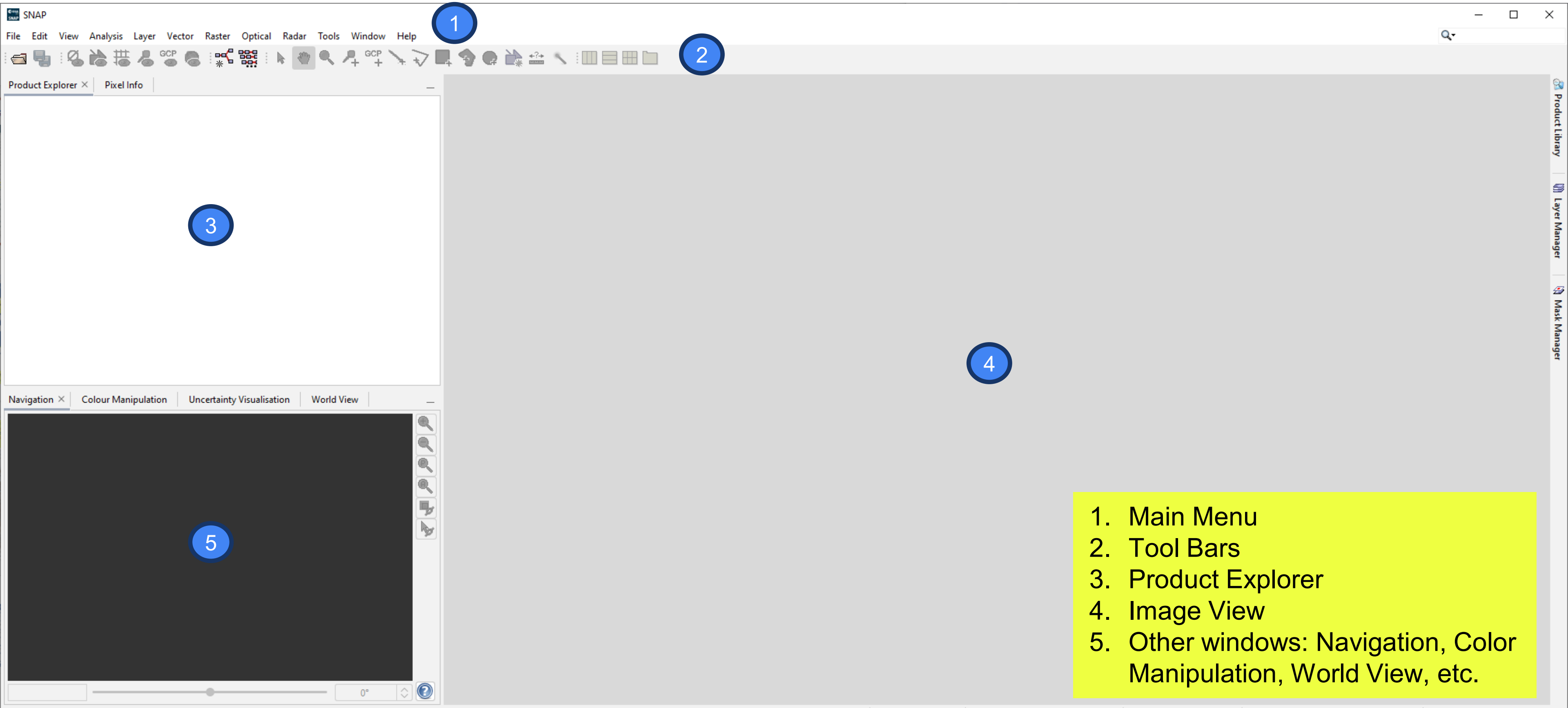
## Urban flood mapping using Sentinel-1 coherence changes

**Processing using Google Earth Engine (GEE)**



# Data Exploration in SNAP

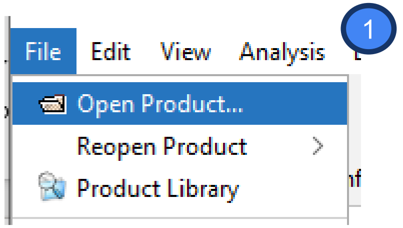
## Open the SNAP software

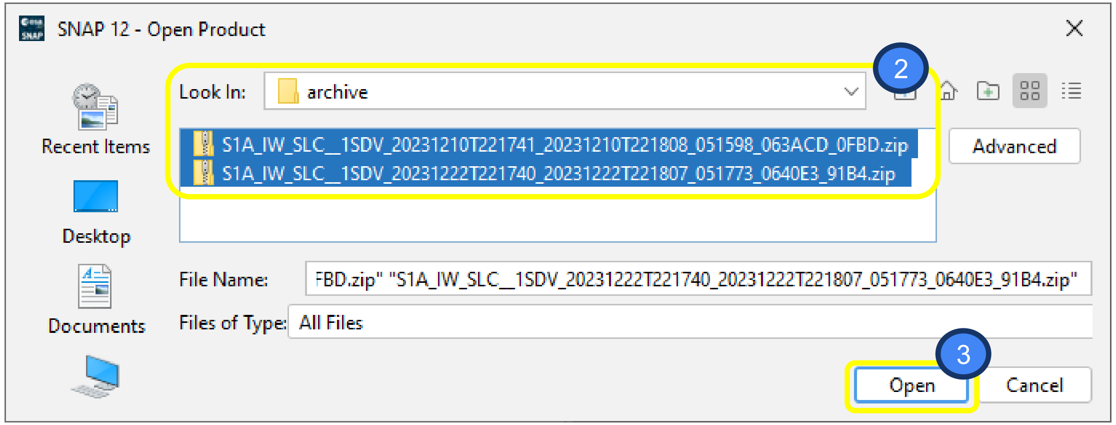


1. Main Menu
2. Tool Bars
3. Product Explorer
4. Image View
5. Other windows: Navigation, Color Manipulation, World View, etc.

# Data Exploration in SNAP

## Open Sentinel-1 Data




SNAP 12 - Open Product

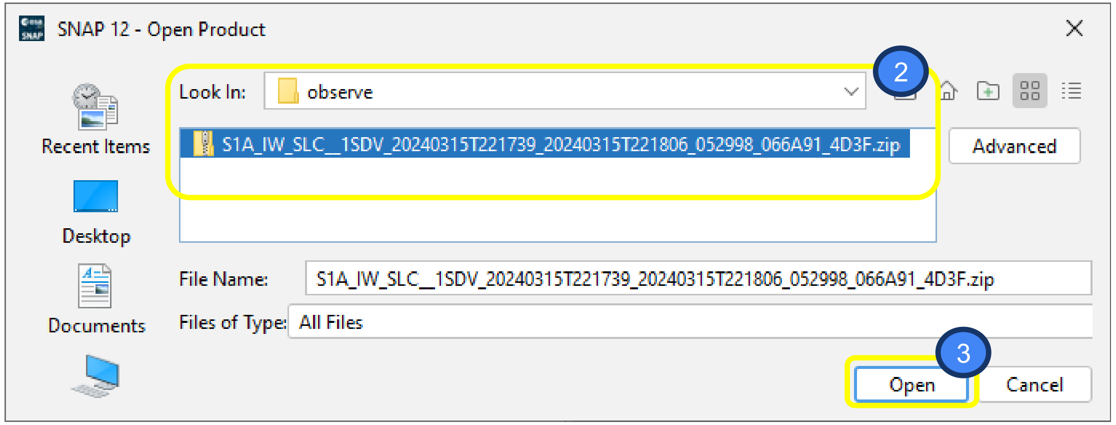
Look In: archive

S1A_IW_SLC__1SDV_20231210T221741_20231210T221808_051598_063ACD_0FBD.zip

S1A_IW_SLC__1SDV_20231222T221740_20231222T221807_051773_0640E3_91B4.zip

File Name: FBD.zip" "S1A_IW_SLC__1SDV_20231222T221740_20231222T221807_051773_0640E3_91B4.zip"

Files of Type: All Files


SNAP 12 - Open Product

Look In: observe

S1A_IW_SLC__1SDV_20240315T221739_20240315T221806_052998_066A91_4D3F.zip

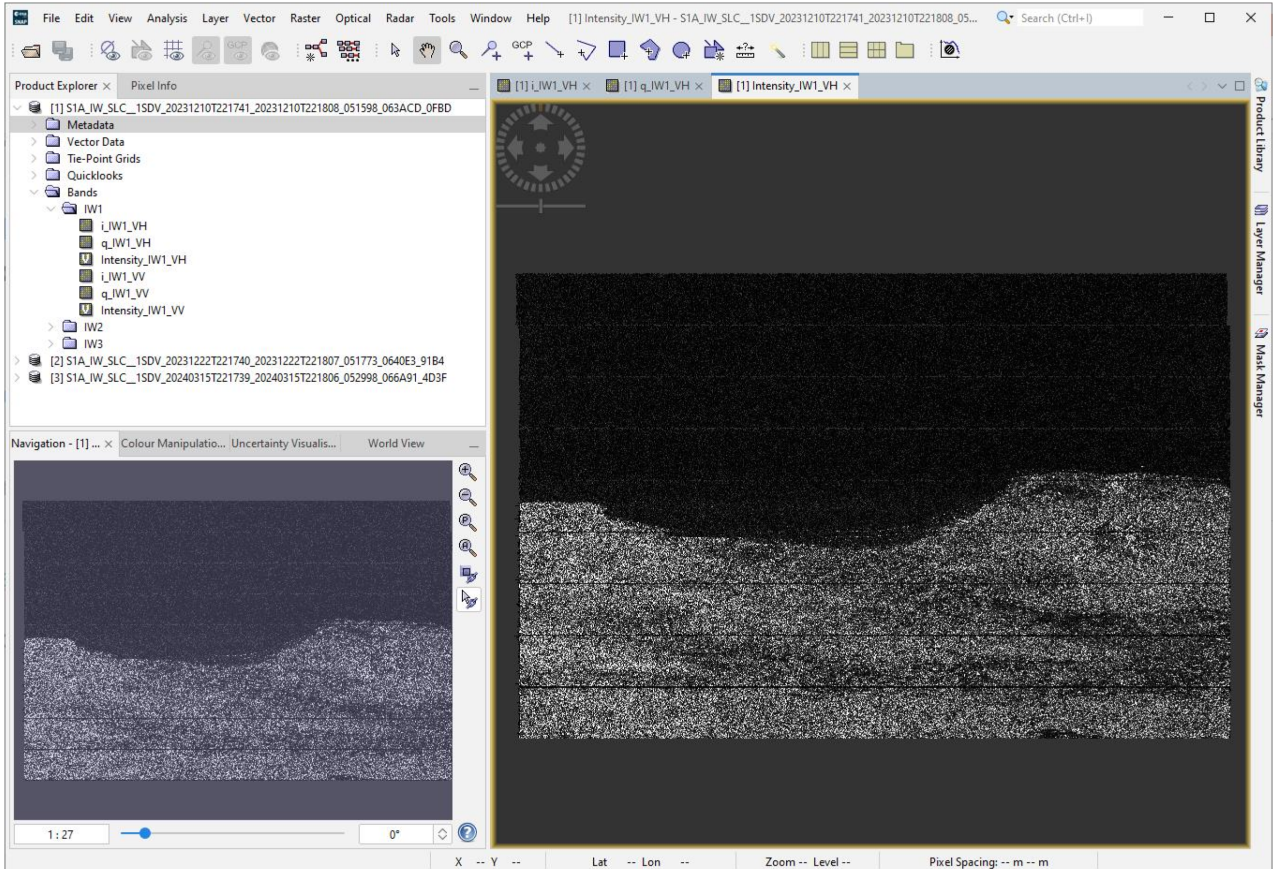
File Name: S1A_IW_SLC__1SDV_20240315T221739_20240315T221806_052998_066A91_4D3F.zip

Files of Type: All Files

1. In the Main Menu, go to File → Open Product…
2. Browse to the location of the data. Then select all Sentinel-1 data in both archive and observe folders. Each file refers to a different acquisition date.
3. Click Open

# Data Exploration in SNAP

## Explore the Sentinel-1 Data



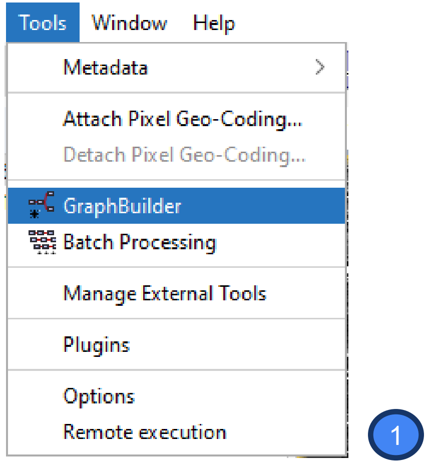
The opened products will appear in the Product Explorer window.

1. Click > to expand the contents of the product [1], then expand the Bands folder. You can see three sub-swaths (IW1, IW2, IW3). Expand one of the folder and you will see the complex data and intensity for each VH and VV polarizations.
2. Double-click on the i_IW1_VH, q_IW1_VH (complex data), and Intensity_IW1_VH band to visualize it. Notice there are 9 burst for each sub-swath.

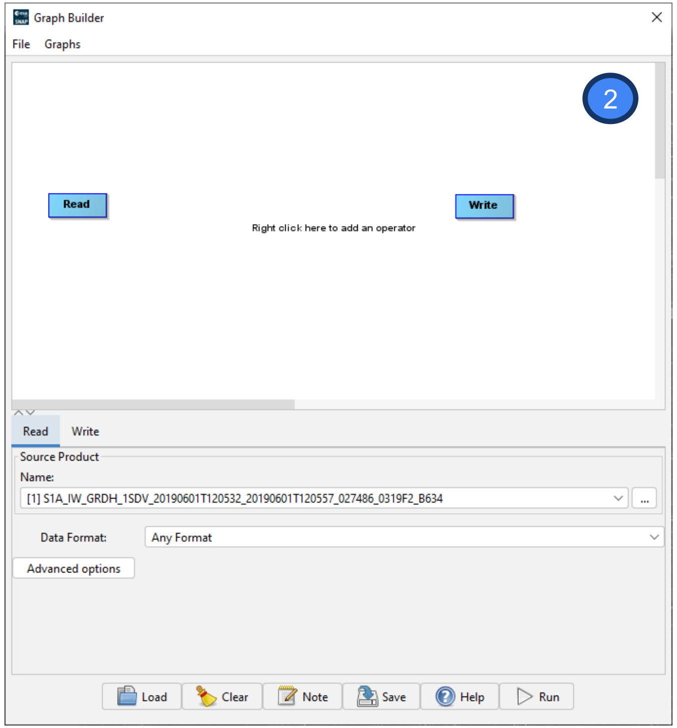
# Data Processing in SNAP

## Open Graph Builder

We will use the **Batch Processing** tool available in SNAP to apply all steps to both images in one go (this also saves disk space as only the final product is physically saved).



1. In the Main Menu, go to Tools → GraphBuilder



[1] S1A_IW_GRDH_1SDV_20190601T120532_20190601T120557_027486_0319F2_B634

2. The **Graph Builder** window will show up.

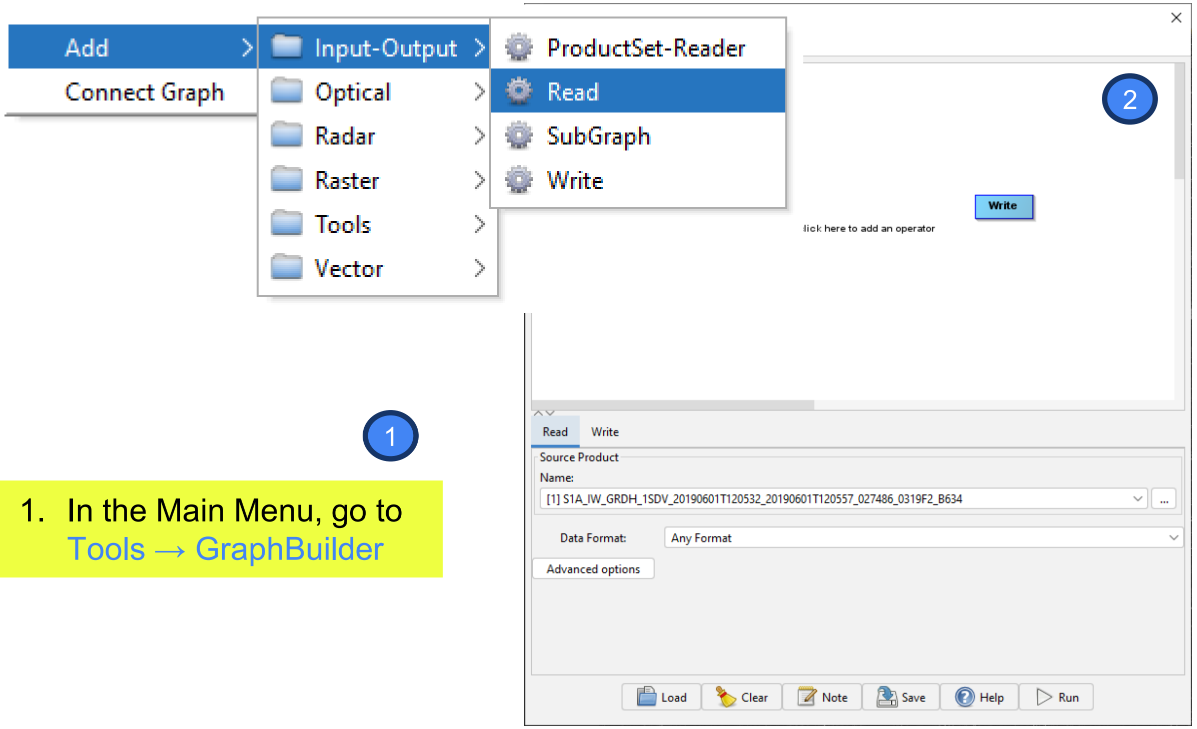
In the beginning, the graph has only two operators: **Read** (to read the input) and **Write** (to write the output). We will create a step-by-step workflow to apply identical pre-processing steps to both of our scenes.

You can delete the **Write** operator for now (Right-click on the operator and select Delete).

# Data Processing in SNAP

## Prepare Read operators

We will create two more Read operators in addition to the default one because we have three input data.



Source Product
Name: [1] S1A_IW_GRDH_1SDV_20190601T120532_20190601T120557_027486_0319F2_B634
Data Format: Any Format
Advanced options

1. In the Main Menu, go to Tools → GraphBuilder

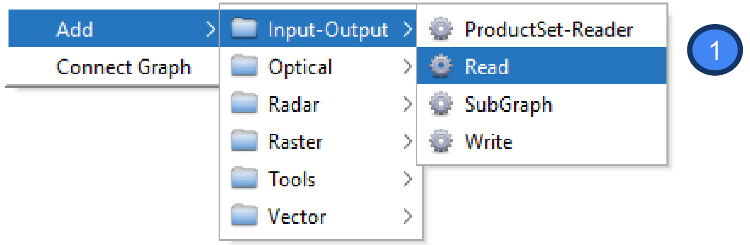
2. The **Graph Builder** window will show up.

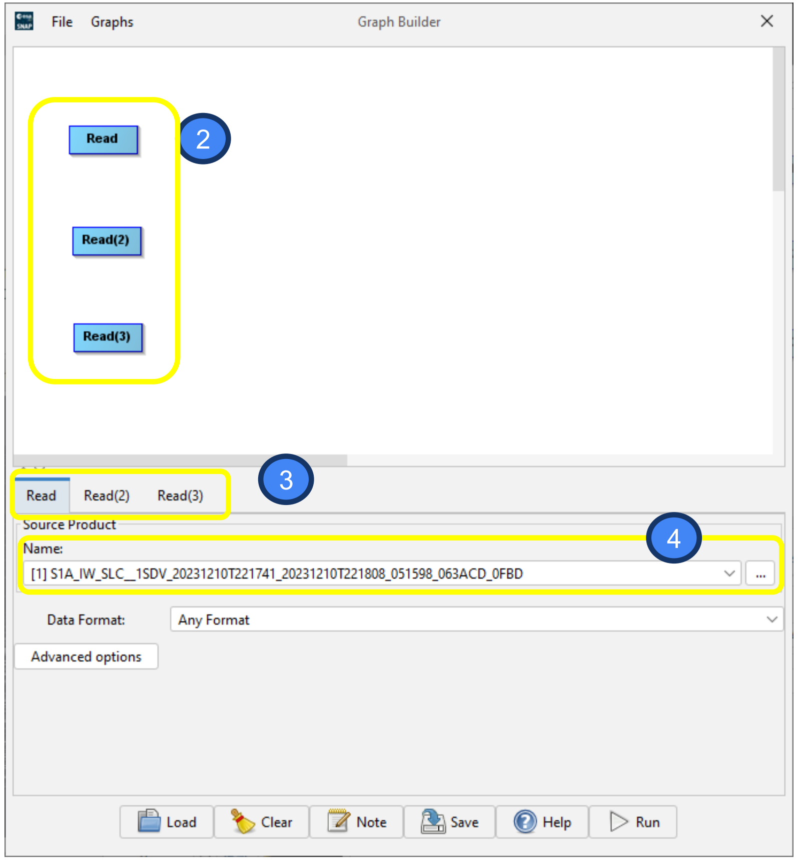
In the beginning, the graph has only two operators: **Read** (to read the input) and **Write** (to write the output). We will create a step-by-step workflow to apply identical pre-processing steps to both of our scenes.

You can delete the **Write** operator for now (Right-click on the operator and select Delete).

# Data Processing in SNAP

## Prepare Read operators





Read | Read(2) | Read(3)

Source Product

Name: [1] S1A_IW_SLC__1SDV_20231210T221741_20231210T221808_051598_063ACD_0FBD

Data Format: Any Format

Advanced options

1. To add the operator right-click the white space between the existing operators and go to Add → Input-Output → Read
2. A new **Read** operator rectangle appeared in our graph. Add one more **Read** operator.
3. Notice that a new tab also appeared below the graph.
4. In each **Read** tabs, select the scenes in the right order.
   - **Read** = Pre-event 1 (20231210)
   - **Read(2)** = Post-event (20241222)
   - **Read(3)** = Pre-event 2 (20231222)

## Data Processing in SNAP

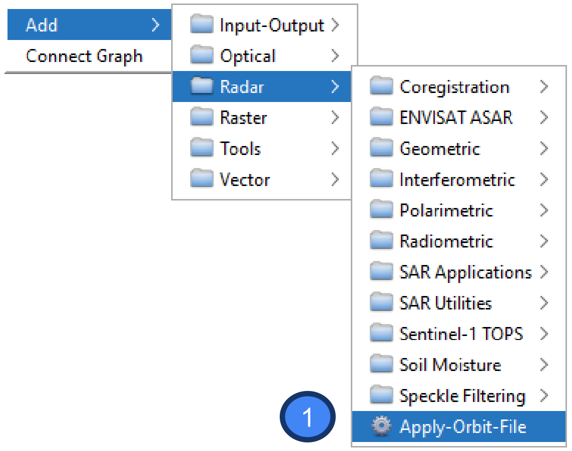
### Update the orbit metadata

- The orbit state vectors provided in the metadata of a SAR product are generally not accurate and can be refined with the precise orbit files which are available days-to-weeks after the generation of the product.

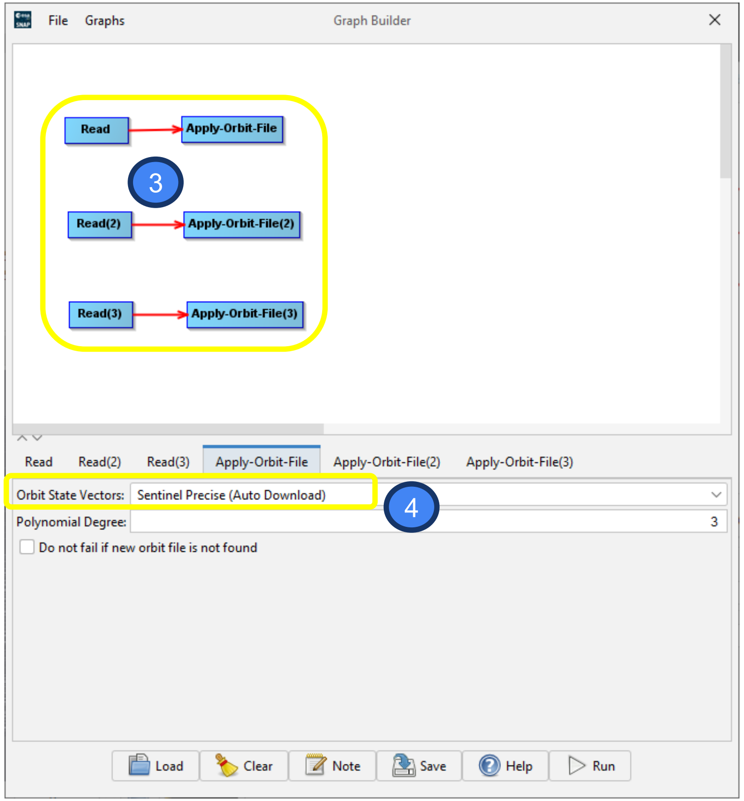
- The orbit file provides accurate satellite position and velocity information. Based on this information, the orbit state vectors in the abstract metadata of the product are updated.

# Data Processing in SNAP

## Update the orbit metadata



1. Right-click the white space between the existing operators and go to Add → Radar → Apply-Orbit-File.



Orbit State Vectors: Sentinel Precise (Auto Download)
Polynomial Degree: 3
Do not fail if new orbit file is not found

3. Now connect the new **Apply-Orbit-File** operator with the Read operator by clicking to the right side of the Read operator and dragging the red arrow towards the **Apply-Orbit-File** operator. Do the same to all.
4. Check that the new tabs appear below the graph. Choose Sentinel Precise (Auto Download).

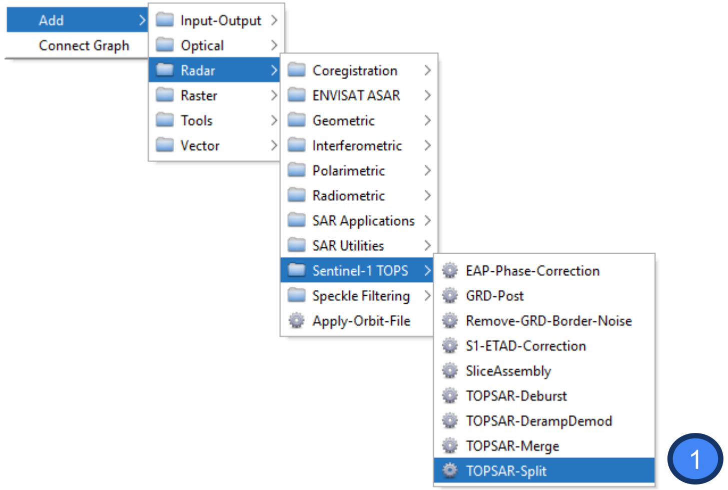
# Data Processing in SNAP

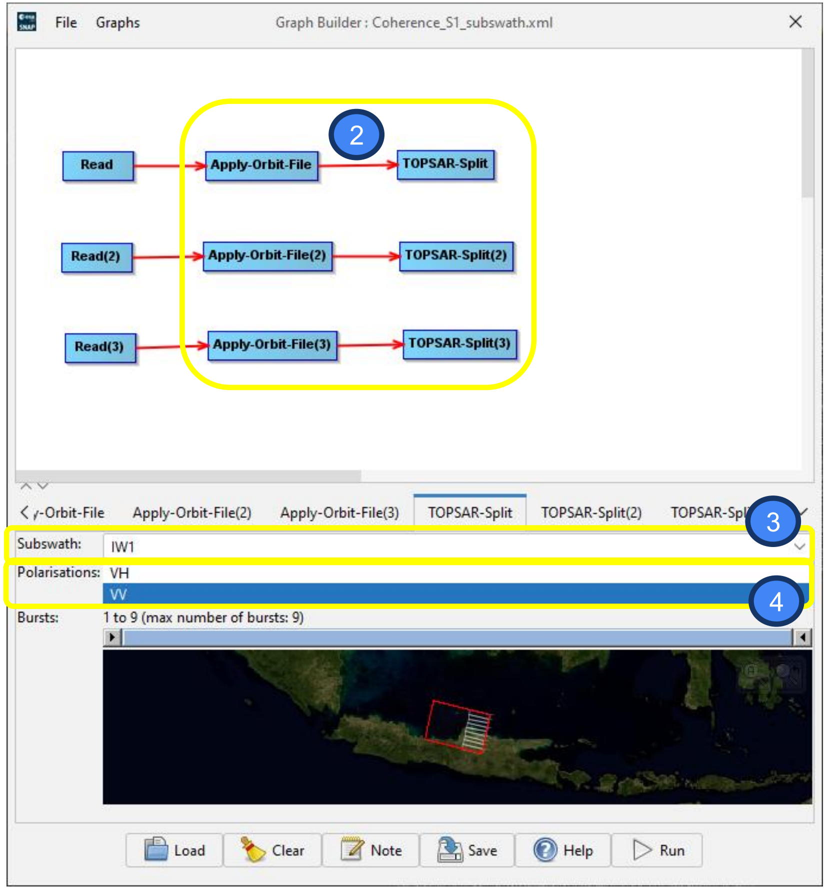
## TOPSAR Split

- Sentinel-1 SLC images in Interferometric Wide (IW) mode typically consist of 3 sub-swaths (IW1, IW2, IW3) — each composed of multiple bursts.
- The TOPSAR Split operator provides a convenient way to split each sub-swath with selected bursts into a separate product.
- The user may select the desired sub-swath with desired bursts and polarizations.

# Data Processing in SNAP

## TOPSAR Split





Tabs: y-Orbit-File, Apply-Orbit-File(2), Apply-Orbit-File(3), TOPSAR-Split, TOPSAR-Split(2), TOPSAR-Spl

Subswath: IW1

Polarisations: VH, VV

Bursts: 1 to 9 (max number of bursts: 9)

1. Right-click the white space between the existing operators and go to Add → Radar → Sentinel-1 TOPS → TOPSAR-Split.
2. Connect the new operator with the Apply-Orbit-File operators.
3. Select the Subswath: IW1. The positioning of the subsets is visible in the bottom map.
4. Select VV polarization.

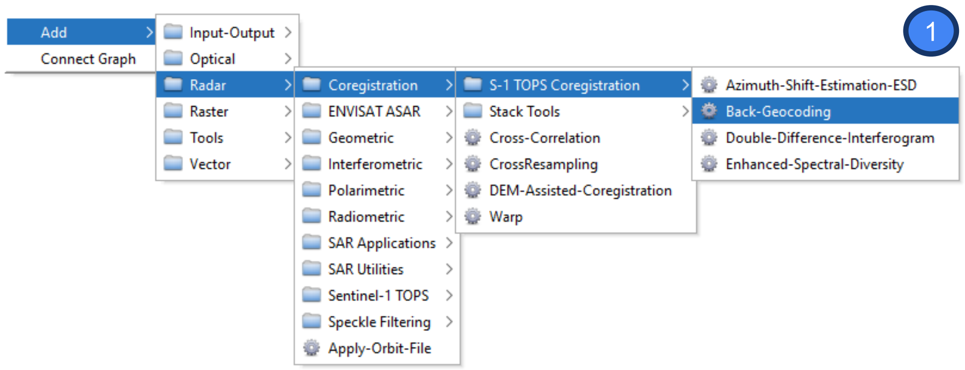
## Data Processing in SNAP

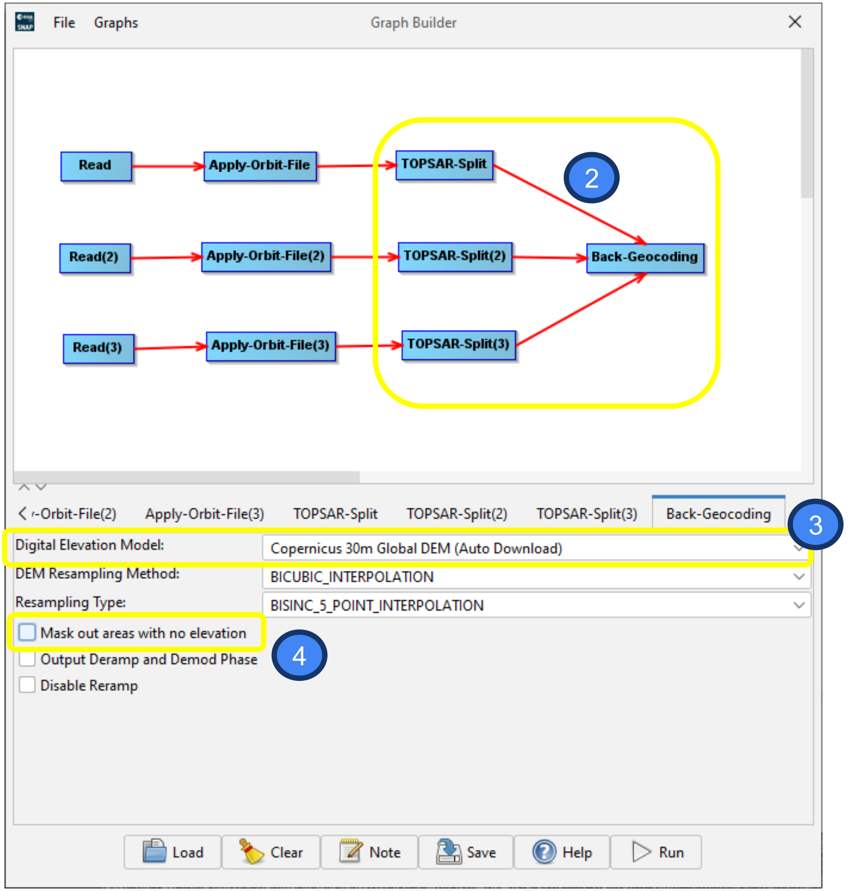
### Back Geocoding

- Due to satellite movement, orbital drift, and acquisition differences, reference and secondary SLC images need to be geometrically matched.
- This operator co-registers two S-1 SLC split products (reference and secondary) of the same sub-swath using the orbits of the two products and a Digital Elevation Model (DEM).

# Data Processing in SNAP

## Back Geocoding





Digital Elevation Model: Copernicus 30m Global DEM (Auto Download)
DEM Resampling Method: BICUBIC_INTERPOLATION
Resampling Type: BISINC_5_POINT_INTERPOLATION
☐ Mask out areas with no elevation
☐ Output Deramp and Demod Phase
☐ Disable Reramp

1. Right-click the white space between the existing operators and go to Add → Radar → Coregistration → S1-TOPS Coregistration → Back Geocoding
2. Connect the new operator with all TOPSAR-Split operators.
3. Select DEM: Copernicus 30m.
4. Uncheck "Mask out areas with no elevation". It is recommended to avoid artefacts along the coast in the co-registered images.

# Data Processing in SNAP

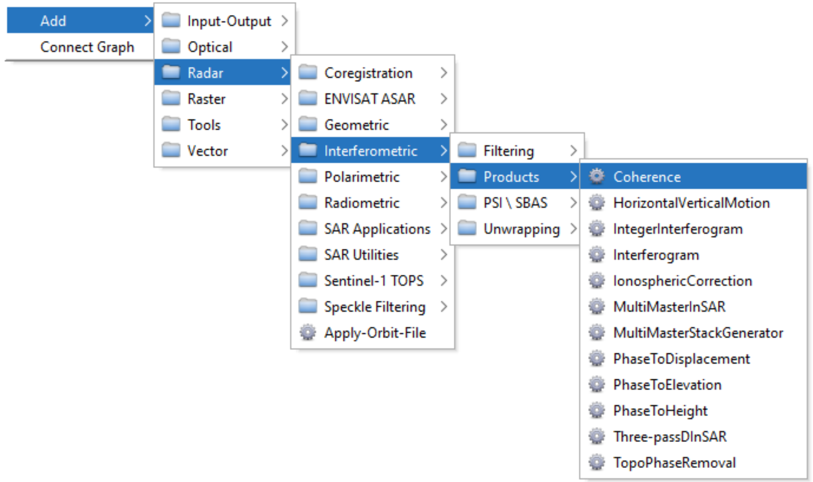
## Coherence Estimation

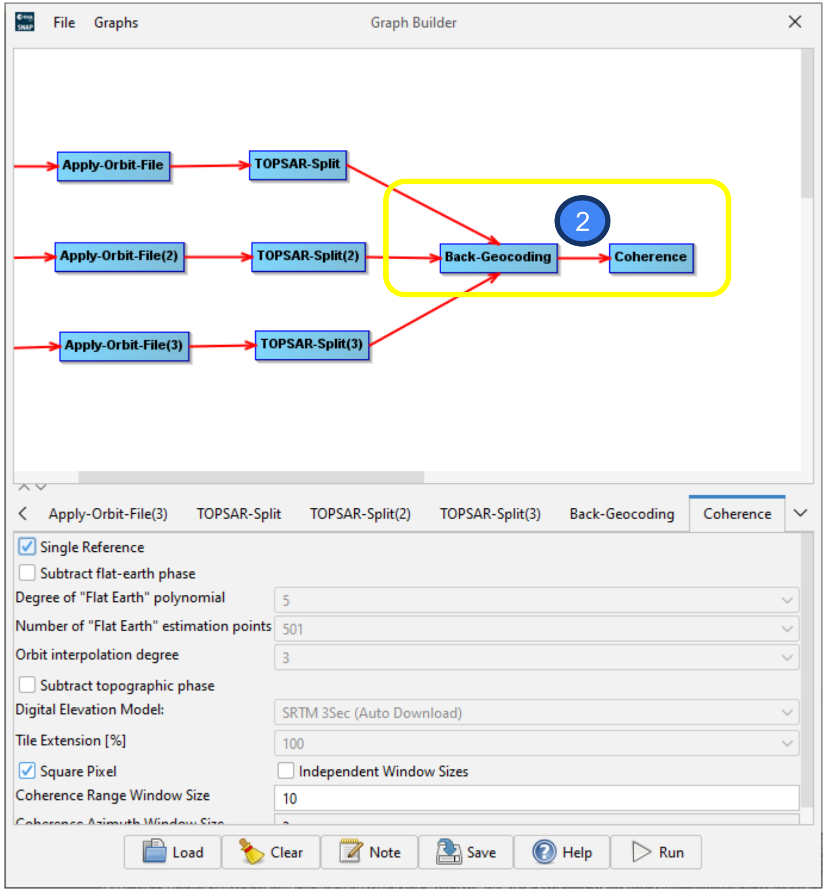
- Coherence is a measure of the similarity in radar signal phase and amplitude between two SAR images acquired at different times, for the same location.

- The Coherence Estimation operator in SNAP calculates the interferometric coherence between a pair of co-registered Sentinel-1 SLC images (reference and secondary).

# Data Processing in SNAP

## Coherence Estimation





- [✓] Single Reference
- [ ] Subtract flat-earth phase
- Degree of "Flat Earth" polynomial: 5
- Number of "Flat Earth" estimation points: 501
- Orbit interpolation degree: 3
- [ ] Subtract topographic phase
- Digital Elevation Model: SRTM 3Sec (Auto Download)
- Tile Extension [%]: 100
- [✓] Square Pixel
- [ ] Independent Window Sizes
- Coherence Range Window Size: 10
- Coherence Azimuth Window Size

1. Right-click the white space between the existing operators and go to Add → Radar → Interferometric → Products → Coherence
2. Connect the new operator with Back-Geocoding operator.
3. Keep the default settings.

## Data Processing in SNAP

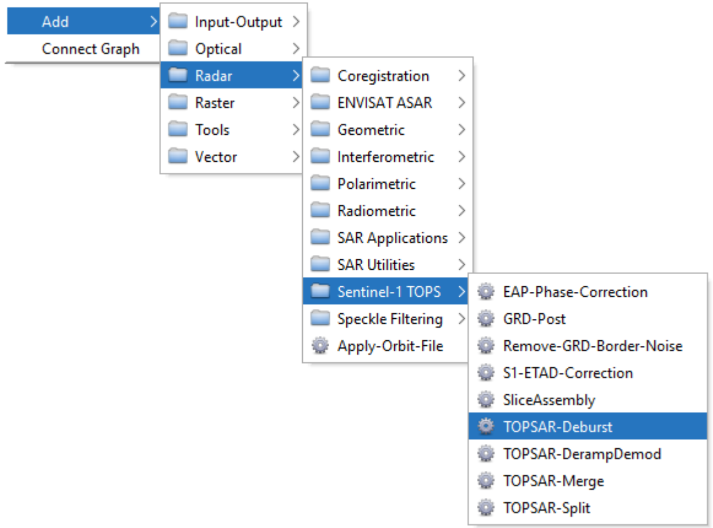
### TOPSAR Deburst

- Each sub-swath consists of a series of adjacent but slightly overlapping bursts, which can cause visible discontinuities (burst gaps) in the interferogram.

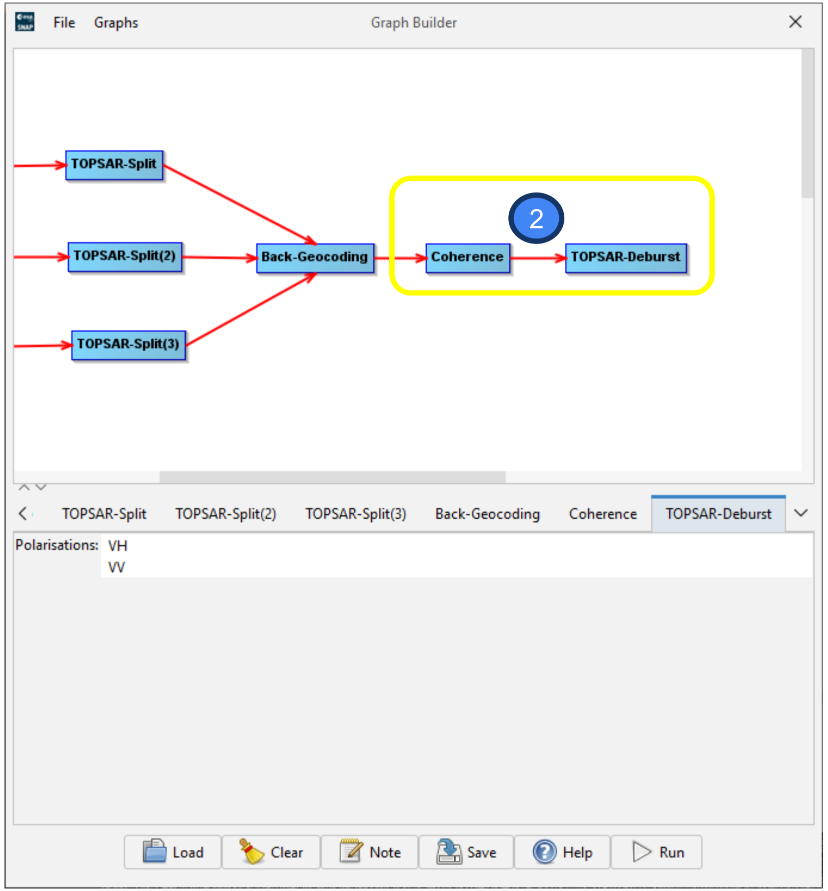
- Debursting removes those discontinuities and stitches the bursts together to create a continuous, seamless image for each sub-swath.

# Data Processing in SNAP

## TOPSAR Deburst



1. Right-click the white space between the existing operators and go to Add → Radar → Sentinel-1 TOPS → TOPSAR-Deburst
2. Connect the new operator with the Coherence operator.
3. Keep the default settings.



Polarisations: VH
VV

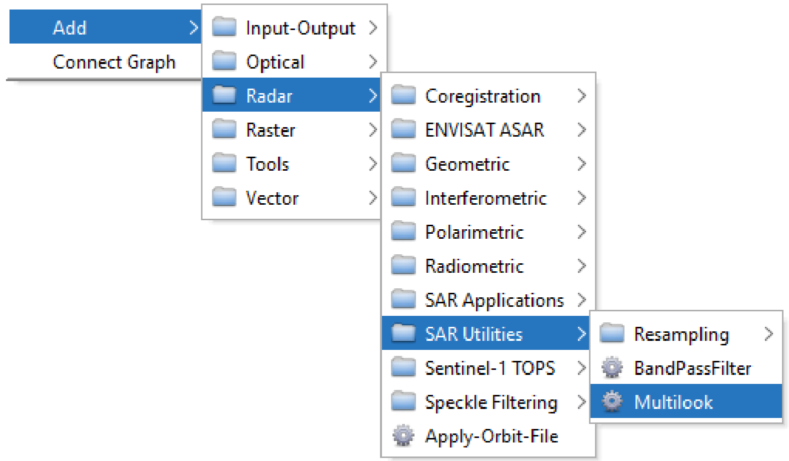
# Data Processing in SNAP

## Multilook

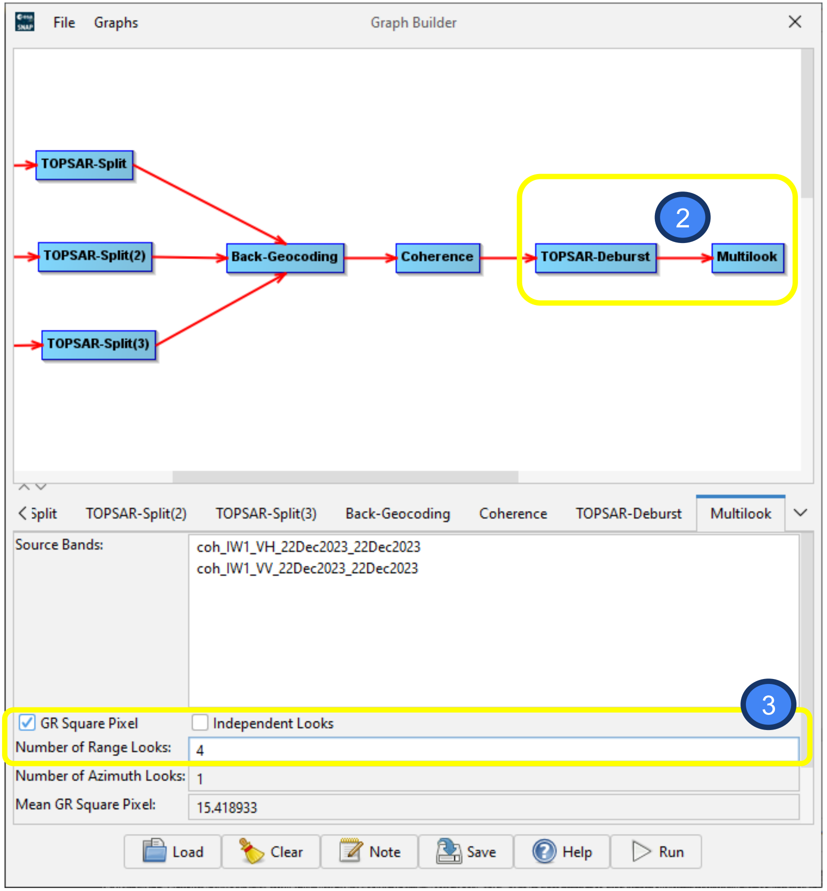
- Multilook is a SAR image processing technique used to reduce speckle noise and improve the visual appearance and radiometric quality of SAR images by averaging neighboring pixels in range and azimuth directions.
- It can be produced by space-domain averaging of a single look image or by a frequency-domain method using the sub-spectral band width.
- Additionally, multi-look processing can be used to reduce the image pixel size.

# Data Processing in SNAP

## Multilook



1. Right-click the white space between the existing operators and go to Add → Radar → SAR Utilities → Multilook
2. Connect the new operator with the TOPSAR-Deburst operator.
3. Specify the Number of Range Looks: 4. If you check the GR Square Pixel option, the range and azimuth spacings are approximately the same in the multilooked image.



Source Bands: coh_IW1_VH_22Dec2023_22Dec2023
coh_IW1_VV_22Dec2023_22Dec2023

[✓] GR Square Pixel [ ] Independent Looks
Number of Range Looks: 4
Number of Azimuth Looks: 1
Mean GR Square Pixel: 15.418933

# Data Processing in SNAP

## Terrain Correction

- Due to topographical variations of a scene and the tilt of the satellite sensor, distances can be distorted in the SAR images.

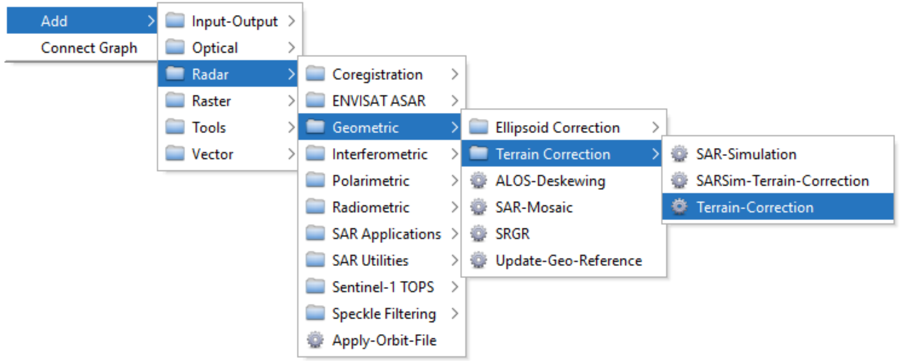
- Image data not directly at the sensor’s Nadir location will have some distortion.

- Terrain corrections are intended to compensate for these distortions so that the geometric representation of the image will be as close as possible to the real world.

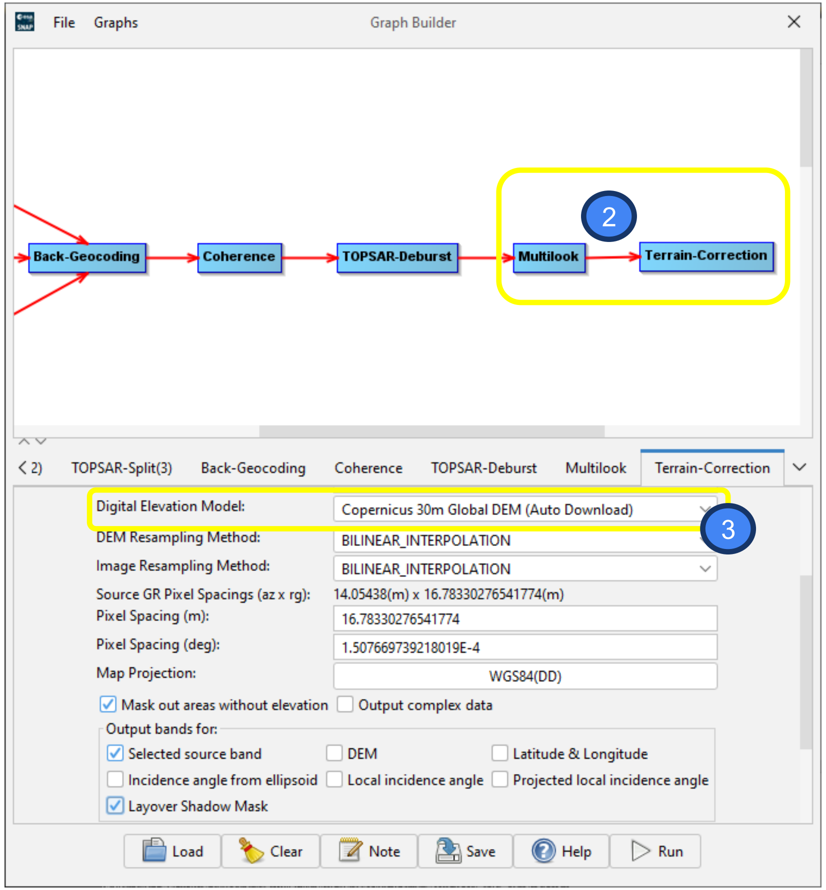
- The Range Doppler Terrain Correction Operator implements the Range Doppler orthorectification method for geocoding SAR images from a single 2D raster radar geometry.

# Data Processing in SNAP

## Terrain Correction



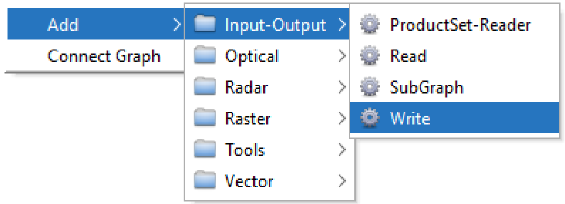
1. Right-click the white space between the existing operators and go to Add → Radar → Geometric → Terrain Correction → Terrain-Correction
2. Connect the new operator with the Multilook operator.
3. Select DEM: Copernicus 30m.



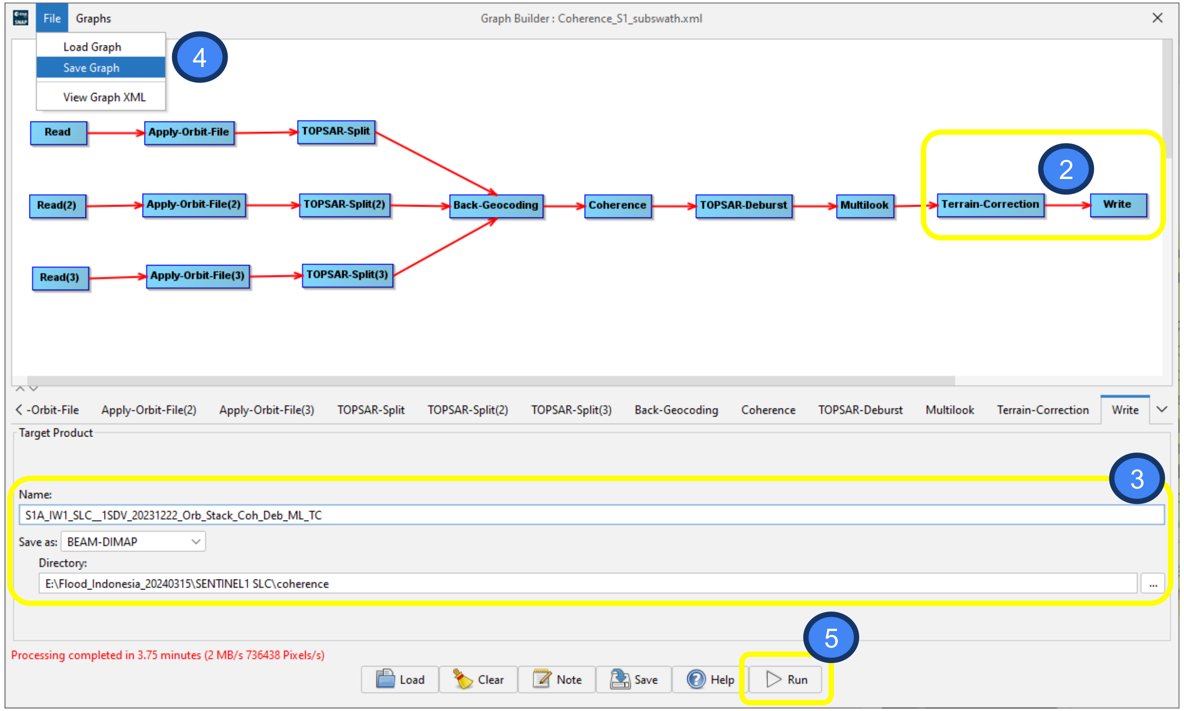
Digital Elevation Model: Copernicus 30m Global DEM (Auto Download)
DEM Resampling Method: BILINEAR_INTERPOLATION
Image Resampling Method: BILINEAR_INTERPOLATION
Source GR Pixel Spacings (az x rg): 14.05438(m) x 16.78330276541774(m)
Pixel Spacing (m): 16.78330276541774
Pixel Spacing (deg): 1.507669739218019E-4
Map Projection: WGS84(DD)
[✓] Mask out areas without elevation [ ] Output complex data
Output bands for:
[✓] Selected source band [ ] DEM [ ] Latitude & Longitude
[ ] Incidence angle from ellipsoid [ ] Local incidence angle [ ] Projected local incidence angle
[✓] Layover Shadow Mask

# Data Processing in SNAP

## Terrain Correction



1. Let's complete the graph with the output. Right-click the white space between the existing operators and go to Add → Input-Output → Write
2. Connect the new operator with the Terrain-Correction operator.
3. Save and change the output directory: S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC
4. Save the graph, go to File → Save Graph in the Graph Builder Main Menu: Coherence_S1_subswath.xml
5. Now that all settings are completed. Run the Graph.



Target Product

Name: S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC

Save as: BEAM-DIMAP

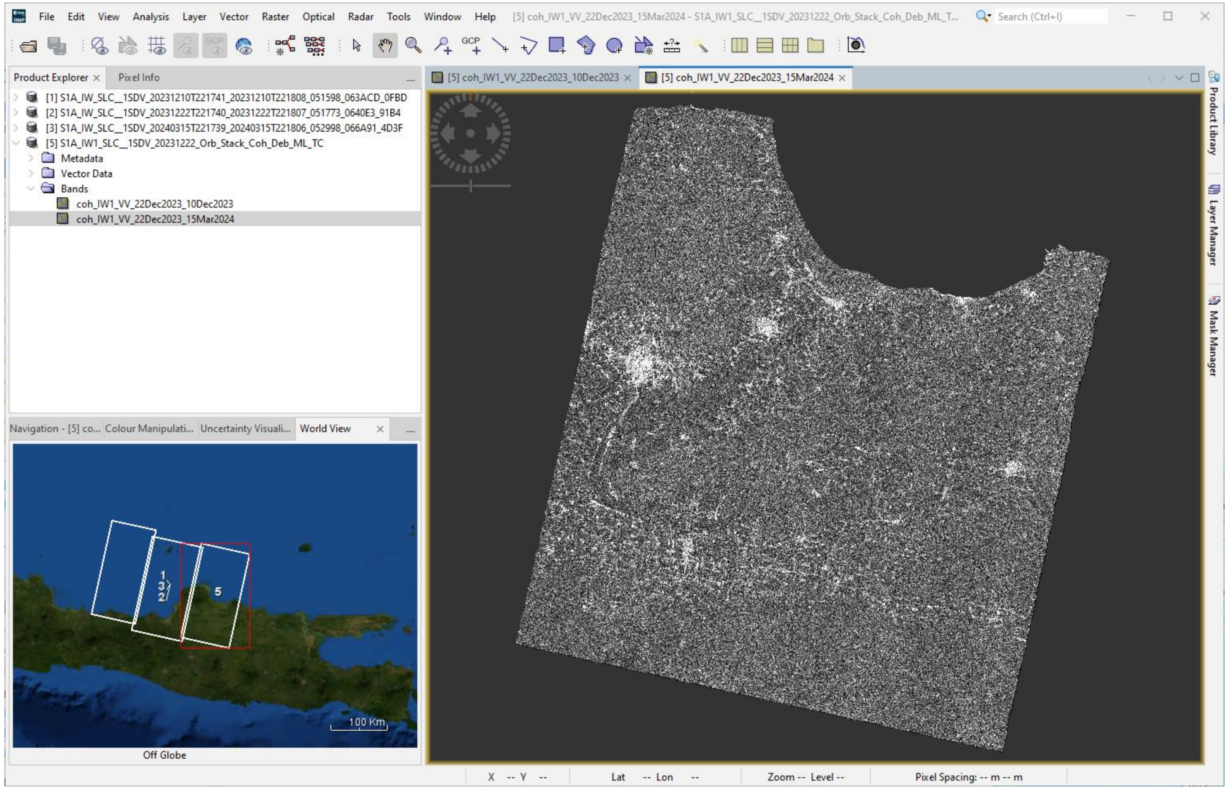
Directory: E:\Flood_Indonesia_20240315\SENTINEL1 SLC\coherence

Processing completed in 3.75 minutes (2 MB/s 736438 Pixels/s)

Here, the processing completed in ~3-4 minutes (depending on your PC/laptop)

# Data Processing in SNAP

## Output: Coherence



[1] S1A_IW_SLC__1SDV_20231210T221741_20231210T221808_051598_063ACD_0FBD
[2] S1A_IW_SLC__1SDV_20231222T221740_20231222T221807_051773_0640E3_91B4
[3] S1A_IW_SLC__1SDV_20240315T221739_20240315T221806_052998_066A91_4D3F
[5] S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC
Metadata
Vector Data
Bands
coh_IW1_VV_22Dec2023_10Dec2023
coh_IW1_VV_22Dec2023_15Mar2024

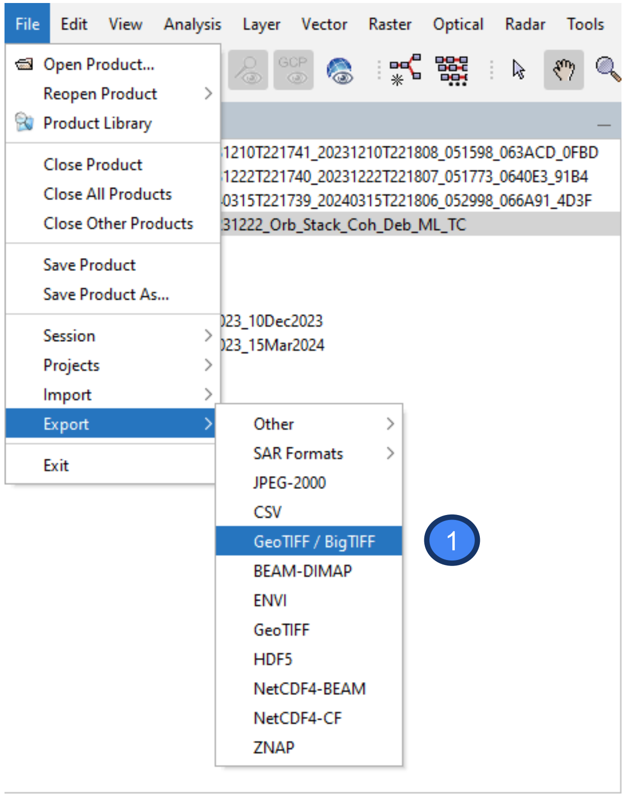
The coherence products will appear in the Product Explorer window.

1. Click > to expand the contents of the product [1], then expand the Bands folder. You can see the coherence of pre-event (t2-t1) and post-event (t2-t3). Make sure the combinations are correct.
2. Double-click on both coherence bands to visualize them.

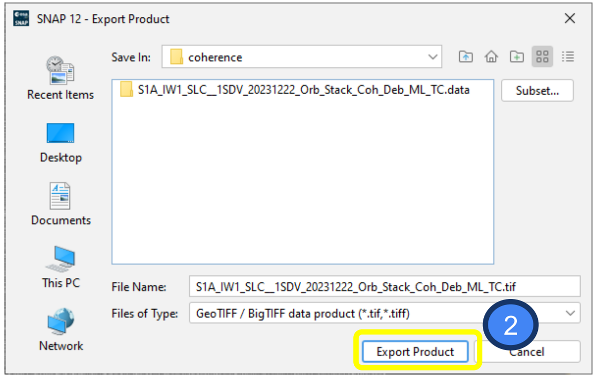
Note: If you have time, you can run the Graph again to process coherence for sub-swath IW2. This sub-swath covers the Semarang area.

# Data Processing in SNAP

## Save the output to GeoTIFF



1210T221741_20231210T221808_051598_063ACD_0FBD
1222T221740_20231222T221807_051773_0640E3_91B4
0315T221739_20240315T221806_052998_066A91_4D3F
31222_Orb_Stack_Coh_Deb_ML_TC
)23_10Dec2023
)23_15Mar2024



Save In: coherence

S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC.data

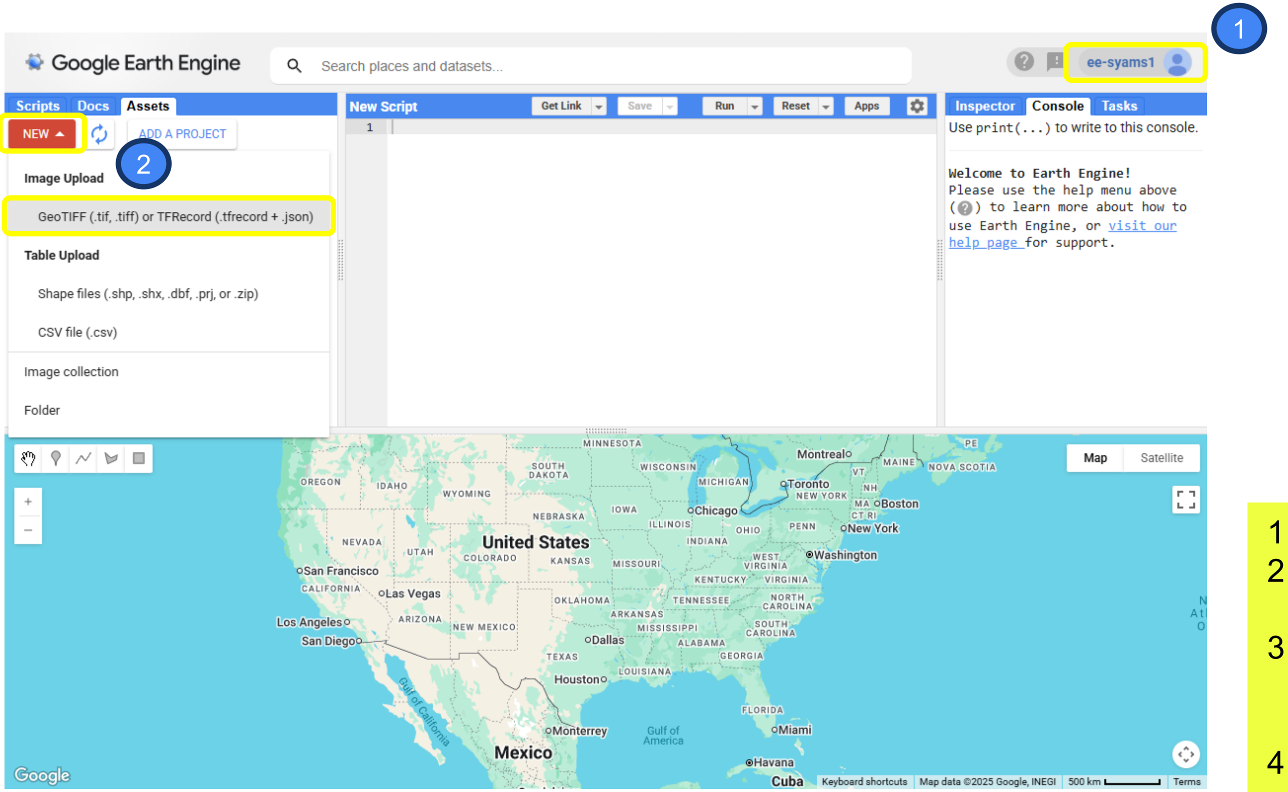
File Name: S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC.tif

Files of Type: GeoTIFF / BigTIFF data product (*.tif,*.tiff)

1. While the output layer is selected, go to File → Export → GeoTIFF/BigTIFF.
2. Click Export Product to save the output.

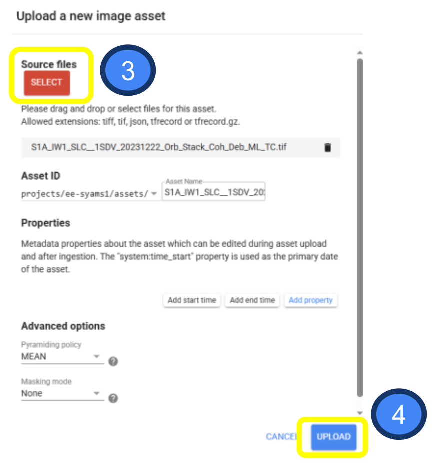
## Data Processing in Google Earth Engine (GEE)

### Load the coherence product to the GEE assets



Use print(...) to write to this console.

```
Welcome to Earth Engine!
Please use the help menu above
(?) to learn more about how to
use Earth Engine, or visit our
help page for support.
```



Upload a new image asset

Source files

Please drag and drop or select files for this asset.
Allowed extensions: tiff, tif, json, tfrecord or tfrecord.gz.

S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC.tif

Asset ID

projects/ee-syams1/assets/ — Asset Name: S1A_IW1_SLC__1SDV_20

Properties

Metadata properties about the asset which can be edited during asset upload and after ingestion. The "system:time_start" property is used as the primary date of the asset.

Advanced options

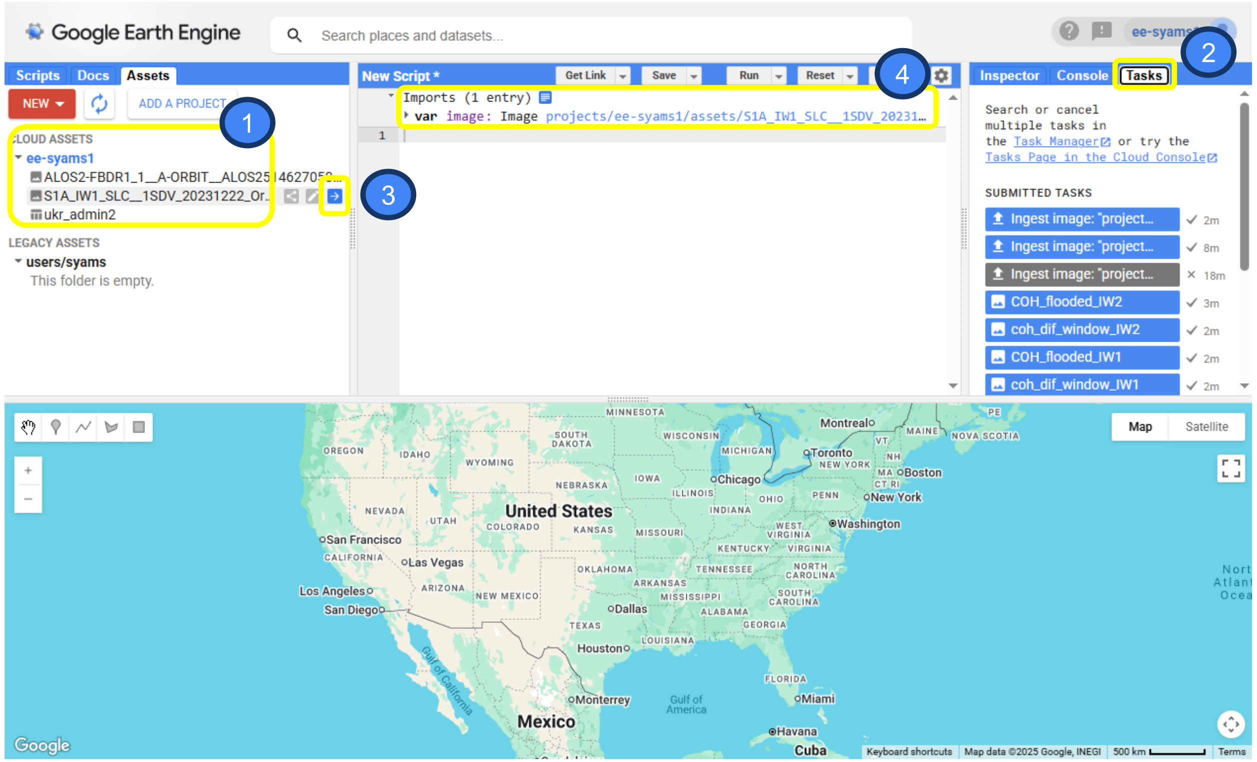
Pyramiding policy: MEAN

Masking mode: None

1. Login to your GEE account.
2. Go to the Assets tab, then click NEW and GeoTIFF.
3. Click SELECT and upload the GeoTIFF coherence product from SNAP.
4. Click UPLOAD.

# Data Processing in Google Earth Engine (GEE)

## Import the coherence product to the GEE script window



```
Imports (1 entry)
 var image: Image projects/ee-syams1/assets/S1A_IW1_SLC__1SDV_20231...
```

Search or cancel multiple tasks in the Task Manager or try the Tasks Page in the Cloud Console

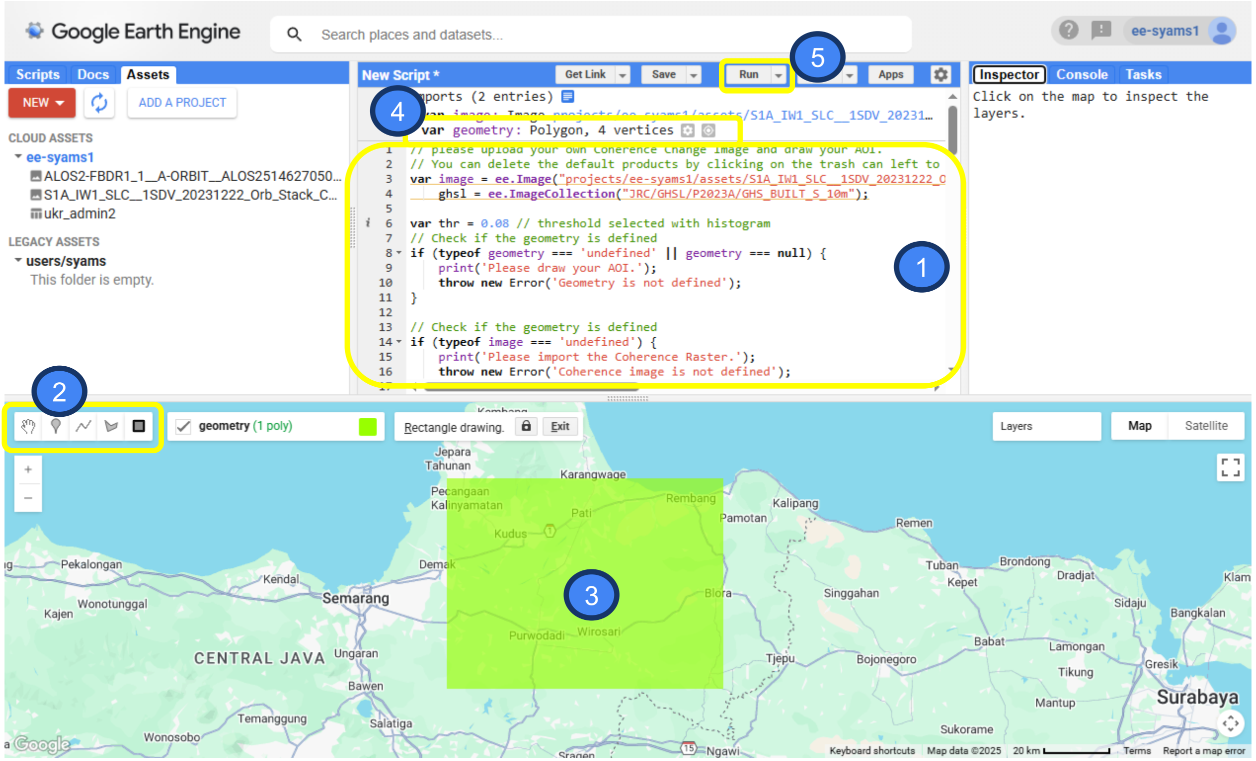
SUBMITTED TASKS

| | | |
|---|---|---|
| Ingest image: "project... | ✓ | 2m |
| Ingest image: "project... | ✓ | 8m |
| Ingest image: "project... | ✕ | 18m |
| COH_flooded_IW2 | ✓ | 3m |
| coh_dif_window_IW2 | ✓ | 2m |
| COH_flooded_IW1 | ✓ | 2m |
| coh_dif_window_IW1 | ✓ | 2m |

1. Once uploading is complete, it will be available as **Cloud Assets**. You may have to click restart to see the product on the **Assets** tab.
2. Check the **Tasks** tab to check the uploading status.
3. Import into the script by clicking
4. You see the image imported into the script window.

# Data Processing in Google Earth Engine (GEE)

## Copy the script and create an AOI



```
Imports (2 entries)
  var image: Image projects/ee-syams1/assets/S1A_IW1_SLC__1SDV_20231…
  var geometry: Polygon, 4 vertices
```

```
// please upload your own Coherence Change image and draw your AOI.
// You can delete the default products by clicking on the trash can left to
var image = ee.Image("projects/ee-syams1/assets/S1A_IW1_SLC__1SDV_20231222_O
    ghsl = ee.ImageCollection("JRC/GHSL/P2023A/GHS_BUILT_S_10m");

var thr = 0.08 // threshold selected with histogram
// Check if the geometry is defined
if (typeof geometry === 'undefined' || geometry === null) {
    print('Please draw your AOI.');
    throw new Error('Geometry is not defined');
}

// Check if the geometry is defined
if (typeof image === 'undefined') {
    print('Please import the Coherence Raster.');
    throw new Error('Coherence image is not defined');
```

1. Copy the prepared script to the Script window.
2. We can create the area of interest using the tools available on the top-left Map window.
3. Draw a polygon on the Map window.
4. The created area of interest will be visible in the Script window.
5. Run the script.

# Data Processing in Google Earth Engine (GEE)

## Script description

```
// please upload your own Coherence Change Image and draw your AOI.
// You can delete the default products by clicking on the trash can left to the import.
var image = ee.Image("projects/ee-syams1/assets/S1A_IW1_SLC__1SDV_20231222_Orb_Stack_Coh_Deb_ML_TC"),
    ghsl = ee.ImageCollection("JRC/GHSL/P2023A/GHS_BUILT_S_10m");
```

The script imports two datasets:

- A **coherence change image** from Sentinel-1 SAR, uploaded as an Earth Engine asset. Change the path accordingly to the location of your coherence product.
- The **Global Human Settlement Layer (GHSL)** 2023 built-up area dataset at 10-meter resolution from the JRC, representing global built-up surfaces.

```
var thr = 0.08 // threshold selected with histogram
```

This line of code defines a threshold value, which was selected based on inspecting a histogram of the image's pixel values. It will be used later in the script to classify or mask areas, identifying areas with coherence values above 0.08 as changed (flooded) and those below as unchanged. You can change the number accordingly.

# Data Processing in Google Earth Engine (GEE)

## Script description

```
// Check if the geometry is defined
if (typeof geometry === 'undefined' || geometry === null) {
    print('Please draw your AOI.');
    throw new Error('Geometry is not defined');
}

// Check if the geometry is defined
if (typeof image === 'undefined') {
    print('Please import the Coherence Raster.');
    throw new Error('Coherence image is not defined');
}
var built_total = ghsl.select('built_surface');
var ccd_vis = {min: 0, max: 1, bands: ['b1', 'b2', 'b2']}; // RGB Visualization
var builtup_vis = {min: 0.0, max: 100.0, palette: ['000000', 'FFFFFF']};
var populationCountVis = {
  min: 0.0,
  max: 100.0,
  palette:
      ['000004', '320A5A', '781B6C', 'BB3654', 'EC6824', 'FBB41A', 'FCFFA4']
};

Map.addLayer(image, ccd_vis, 'Coherence Change detection', 1);
Map.addLayer(built_total.first().clip(geometry), builtup_vis, 'Built-up surface [m^2], 2018', 0);
```

The script sets up error handling for missing inputs and defines color palettes for displaying raster data layers. It also adds the coherence change image and built-up surface layers to the map display in Earth Engine

## Data Processing in Google Earth Engine (GEE)

### Script description

```
// create Mask from Built up layer with buffer

var built_up = built_total.first().clip(geometry).gte(1);
var builtup_res = built_up.reduceResolution({
    reducer:ee.Reducer.mean(),
    maxPixels: 1024
  })
  .reproject({
    crs: image.projection()
  })

var coh_dif = image.select('b1').subtract(image.select('b2'))
coh_dif = coh_dif.updateMask(builtup_res)

Map.addLayer(coh_dif, {min: -0.3, max: 0.3, palette:['blue', 'white', 'red']}, 'Coherence Difference', 0)

var coh_dif_window = coh_dif.reduceNeighborhood({
  reducer: ee.Reducer.median(), // potentially other statistic
  kernel: ee.Kernel.square(10) // is this in pixels?
});
Map.addLayer(coh_dif_window, {min: -0.4, max: 0.4, palette:['blue', 'white', 'red']}, 'Coherence Difference in Moving window', 1)
```

This script focuses on analyzing coherence differences within built-up areas, visualizing raw differences, and a smoothed version using a moving window to highlight local patterns of change.

## Data Processing in Google Earth Engine (GEE)

### Script description

```
// Plot Value Distribution
var chart =
    ui.Chart.image.histogram({image: coh_dif_window, region: geometry, scale: 50,
    minBucketWidth: 0.002, maxBuckets: 100})
        .setOptions({
          title: 'Histogram',
          hAxis: {
            title: 'Coherence Difference',
            titleTextStyle: {italic: false, bold: true},
          },
          vAxis:
              {title: 'Count', titleTextStyle: {italic: false, bold: true}},
          colors: ['cf513e']
        });
print(chart);

var flooded = coh_dif_window.gte(ee.Image.constant(thr))
Map.addLayer(flooded, {min: 0, max: 1, palette:['white', 'blue']}, 'Flooded')
```

This section visualizes the distribution of coherence differences in a histogram and maps areas likely affected by flooding based on a threshold applied to the smoothed coherence difference layer.

# Data Processing in Google Earth Engine (GEE)

## Script description

```
// Mask urban density layer JRC with Flood Layer
// Or some how multiply the two layers
var pop2025 = ee.Image('JRC/GHSL/P2023A/GHS_POP/2020');
// pop2025 = pop2025.clip(geometry).updateMask(flooded)
pop2025 = built_total.first().updateMask(flooded) // higher accuracy in dense urban areas?

Map.addLayer(pop2025, populationCountVis, 'Population count in affected Areas', 0);
```

This section overlays the flood extent on built-up areas to visualize where built-up (and likely populated) areas are affected by flooding — serving as a proxy for estimating population exposure within flooded urban zones.

# Data Processing in Google Earth Engine (GEE)

## Script description

```
// Exports
// Moving Window
Export.image.toDrive({
  image: coh_dif_window,
  description: 'coh_dif_window',
  fileNamePrefix: 'coh_dif_window',
  region: geometry,
  folder: 'UrbanFloods_COH',
  scale: 20,
  maxPixels: 1e13
});
// Moving Window
Export.image.toDrive({
  image: flooded,
  description: 'COH_flooded',
  fileNamePrefix: 'COH_flooded',
  region: geometry,
  folder: 'UrbanFloods_COH',
  scale: 20,
  maxPixels: 1e13
});
```

This final section saves the final processed flood and coherence difference results as GeoTIFF images to Google Drive for external use.

THANK YOU

Geoinformatics Center, Asian Institute of Technology