SenSARP Documentation

Release 0.1

Thomas Weiß

Jan 31, 2022

CONTENTS

1 Content **2**

1.1 Introduction . . . 2
1.2 Installation for Linux (tested with Ubuntu 20.04) . . . 3
1.3 Usage . . . 5
1.4 Default Pre-Processing Chain . . . 42
1.5 Technical documentation . . . 51
1.6 Authors . . . 54
1.7 Change Log SenSARP . . . 54
1.8 Licence . . . 55

2 Credits **65**

Bibliography **66**

Python Module Index **68**

Index **69**

CHAPTER

ONE

CONTENT

1.1 Introduction

The Sentinel-1 mission consists of two polar-orbiting satellites acquiring Synthetic Aperture Radar data (SAR) at C-band (frequency of 5.405 GHz) with a revisit time of 6 days. The SAR data is distributed free of charge via the Copernicus Open Access Hub (https://scihub.copernicus.eu/) by European Space Agency (ESA) and the European Commission. Large archives are also provided by Data and Information Access Services (DIAS) which serve the purpose to facilitate the access and use of Sentinel Data. Due to the specific imaging geometry of the radar system, the acquired radar data contains different radiometric and geometric distortions. The radiometric quality is affected by spreading loss effect, the non-uniform antenna pattern, possible gain changes, saturation, and speckle noise. Geometric distortions such as foreshortening, layover or shadowing effects are based on the side looking radar acquisition system. To account for these radiometric and geometric distortions, the Sentinel-1 Level 1 data has to be corrected radiometrically and geometrically before the data can be used for further analysis or within third party applications. Therefore, either an automatic or manual pre-processing of Sentinel-1 images is needed.

1.1.1 Statement of need

Sentinel-1 satellites will provide continuous free available microwave remote sensing data of the entire globe at least until the end of 2030. Furthermore, ESA is not only providing Sentinel satellite images (e.g. Sentinel-1, Sentinel-2, Sentinel-3) but they also developed free open source toolboxes (Sentinel-1, 2, 3 toolboxes) for scientific exploitation. The toolboxes can be accessed and used via the Sentinel Application Platform (SNAP). SNAP offers a graphical interface were expert users can develop different processing schemes and apply them on the satellite images. Although, Sentinel-1 satellite data and a processing software are freely available, the usage of the data is mainly limited to expert users in the field of microwave remote sensing as different pre-processing steps need to be applied before using Sentinel-1 images.

SenSARP was developed to provide a push-button option to easily apply a rigid pre-processing pipeline with sensible defaults to a Sentinel-1 Level 1 SLC time series data as well as single Sentinel-1 Level 1 SLC images. Thus, non-expert users in the field of pre-processing microwave data are able to use radiometric and geometric corrected sigma nought backscatter data for their specific applications. Beside a rigid pre-processing pipeline, SenSARP provides filter options to retrieve only images of a specific year or images that contain a specific area of interest from a stack of downloaded Sentinel-1 data. Furthermore, the default processing scheme of SenSARP can handle if an area of interest is contained in two tiles of the same swath (due to storage reasons data of one Sentinel-1 satellite swath is provided by ESA within different tiles). Additionally, SenSARP checks if within a stack of Sentinel-1 images, one specific image was multiple processed by ESA and uses the newest.

For expert users, SenSARP provides the possibility to automate their pre-processing on a large scale by either modifying the default pre-processing scheme (modification of xml graph pre_processing_step1.xml) or create their own pre-processing scheme (create a new xml graph) with the graph builder of the SNAP software. They can benefit from the filter options, the default pre-processing step 2 (co-registration of images) and the SenSARP functions to stack all

processed and co-registered images within a netCDF file with additional image information e.g. satellite name, relative orbit and orbitdirection.

1.1.2 Getting Started

Please find instructions on how to download and install SenSARP in the *Installation for Linux (tested with Ubuntu 20.04)* section.

1.1.3 Support, contributing and testing

Please contribute using Github Flow. Create a branch, add commits, and open a pull request.

Reporting bugs

If you find a bug in SenSARP, please open an new issue and tag it "bug".

Suggesting enhancements

If you want to suggest a new feature or an improvement of a current feature, you can submit this on the issue tracker and tag it "enhancement".

Testing

The package is currently tested for Python >= 3.6 on Unix-like systems. To run unit tests, execute the following line from the root of the repository:

```
pytest
```

1.2 Installation for Linux (tested with Ubuntu 20.04)

Note: The SenSARP has been developed against Python 3.6. It cannot be guaranteed to work with previous Python versions.

The first step is to clone the latest code and step into the check out directory:

```
git clone https://github.com/multiply-org/sar-pre-processing.git
cd sar-pre-processing
```

1.2.1 Installation with Conda

Download and install Anaconda or Miniconda. Anaconda/Miniconda installation instructions can be found here

To install all required modules, use:

```
conda env create --prefix ./env --file environment.yml
conda activate ./env # activate the environment
```

To install SenSARP into an existing Python environment, use:

```
python setup.py install
```

To install for development, use:

```
python setup.py develop
```

1.2.2 Installation with virtualenv and python

Install system requirements:

```
sudo apt install python3-pip python3-tk python3-virtualenv python3-venv virtualenv
```

Create a virtual environment:

```
virtualenv -p /usr/bin/python3 env
source env/bin/activate # activate the environment
pip install --upgrade pip setuptools # update pip and setuptools
```

To install SenSARP into an existing Python environment, use:

```
python setup.py install
```

To install for development, use:

```
python setup.py develop
```

GDAL package needs to be installed too:

```
sudo apt install gdal-bin libgdal-dev

python -m pip install pygdal=="`gdal-config --version`.*"
```

1.2.3 Further information

Please see the environment file for a list of all installed dependencies during the installation process. Additionally, ESA's SNAP Sentinel-1 Toolbox (Version >8.0.3) has to be installed prerequisite. The Software can be downloaded here. To install the SNAP toolbox, open a terminal window and use:

```
bash esa-snap_sentinel_unix_8_0.sh
```

SenSARP uses only functionalities of the Sentinel-1 Toolbox.

Note: Currently, only SNAP version 8.0 can be downloaded from the website. To update SNAP to a version >8.0.3 please start the SNAP software. You will be asked if you want to search for update. Please search for updates and install all updates. After the updates are installed, you need to restart SNAP to initialize the updates correctly.

SNAP Toolbox need libgfortran for specific operations but currently libgfortran is not installed during the installation process of SNAP, therefore you might use:

```
sudo apt-get install gfortran
```

1.3 Usage

1.3.1 Example 1: Use default processing graph to pre-process single Sentinel-1 Level-1 SLC image

1. Requirements

- Installation of SenSARP
- Installation of ESA's SNAP Toolbox version >8.0.3
  - Currently only SNAP version 8.0 can be downloaded from the ESA website (https://step.esa.int/main/download/snap-download/). To update SNAP to a version >8.0.3 please start the SNAP software. You will be asked if you want to search for update. After the updates are installed you need to restart SNAP to initialize the installed updates.
  - SNAP Toolbox need libgfortran for specific operations but currently libgfortran is not installed during the installation process of SNAP therefore you might use `sudo apt-get install gfortran`
- Sentinel-1 SLC data
  - Instruction how to download Sentinel 1 data are given in Section 2

2. Download sample data from Sentinel Data Hub

Option 1: Download data from Sentinel Data Hub manually or via python package sentinelsat

Create Account (https://scihub.copernicus.eu/dhus/#/self-registration). (**Attention: Problem by using Copernicus Open Access Hub might be that older data is offline and need to be triggered first**). More information can be found at https://scihub.copernicus.eu/userguide/DataRestoration. Instruction to manually download data from Copernicus Open Access Hub can be found at https://blogs.fu-berlin.de/reseda/esa-scihub/. You can also try to download the data via python package sentinelsat

How to use sentinelsat

[1]:
```
# connect to the API
from sentinelsat import SentinelAPI, read_geojson, geojson_to_wkt
from datetime import date
user = 'user'
password = 'password'
# initialize settings
api = SentinelAPI(user, password)
```

Search for available data

[2]:
```
# search by polygon (MNI test site coordinates), time, and SciHub query keywords
footprint = geojson_to_wkt(read_geojson('coordinates_mni.geojson'))
products = api.query(footprint,
                     date=('20210701', '20210702'),
                     platformname='Sentinel-1',
                     producttype='SLC')
print('Following products will be downloaded')
print(api.to_dataframe(products).title.values)

print('These {} product need {} Gb of disk space'.format(len(products), api.get_products_
↪size(products)))
```

```
Following products will be downloaded
['S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_027596_034B44_2DCF']
These 1 product need 7.72 Gb of disk space
```

Start download process (**Attention: might take a while and data will requries some free disk space**)

[3]:
```
# download all results from the search
# files will be downloaded to specified path
import os
path = os.path.expanduser('~/Desktop/data')
try:
    os.makedirs(path)
except: FileExistsError
api.download_all(products, path)
```

```
Downloading products:   0%|          | 0/1 [00:00<?, ?product/s]
```

```
Downloading S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_027596_034B44_2DCF.zip:   0
↪%|          | 0.00/4.6...
```

```
MD5 checksumming:   0%|          | 0.00/4.62G [00:00<?, ?B/s]
```

[3]:
```
ResultTuple(downloaded={'87bc6000-e45b-4e06-9c8e-1db1908db1ca': {'id': '87bc6000-e45b-
↪4e06-9c8e-1db1908db1ca', 'title': 'S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_
↪027596_034B44_2DCF', 'size': 4620218028, 'md5': 'b142da92c874236fbdd6572480b96128',
↪'date': datetime.datetime(2021, 7, 1, 5, 17, 38, 754000), 'footprint': 'POLYGON((14.
↪731479 47.650288,11.295774 48.053822,11.672503 49.676376,15.223297 49.271305,14.731479␣
↪47.650288))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('87bc6000-
↪e45b-4e06-9c8e-1db1908db1ca')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 7, 1, 7, 24, 3, 755000), 'Ingestion Date': datetime.datetime(2021, 7, 1,
↪ 7, 21, 46, 931000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('87bc6000-e45b-4e06-9c8e-1db1908db1ca')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data/S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_027596_
↪034B44_2DCF.zip', 'downloaded_bytes': 4620218028}}, retrieval_triggered={}, failed={})
```

(continues on next page)

(continued from previous page)

Option 2: Manually search and download data from Alaska Satellite Facility (ASF)

You can search for Sentinel-1 data at https://search.asf.alaska.edu/. A NASA EOSDIS Earthdata Login account is required for downloading data and tools from ASF. Registering for an Earthdata Login account is free (https://urs.earthdata.nasa.gov/home). Instructions how to download data from ASF can be found at https://asf.alaska.edu/wp-content/uploads/2019/02/asf_datarecipe_bulk_download_from_vertex_python_script_v1.pdf.

3. Use default processing graph to pre-process a single Sentinel-1 Level-1 SLC image

Set paths for

- input_folder (path to stored Sentinel-1 SLC data (zip files) e.g. "~/Downloads")
- output_folder (path where processed data will be stored e.g. "~/output")
- gpt_loction (gpt is located in the bin folder of your SNAP installation)'

[4]:
```
input_folder = path
output_folder = path
gpt_location = os.path.expanduser('~/snap/bin/gpt')
```

Create config file with information about input folder, output folder and gpt path

[5]:
```
import yaml

with open('sample_config_file.yaml') as stream:
   data = yaml.safe_load(stream)

data['input_folder'] = input_folder
data['output_folder'] = output_folder
data['gpt'] = gpt_location

with open('test_config_file.yaml', 'wb') as stream:
   yaml.safe_dump(data, stream, default_flow_style=False,
                  explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Optional config options for subsetting

[6]:
```
with open('test_config_file.yaml') as stream:
   data = yaml.safe_load(stream)

## Define region of interest
data['region']['lr']['lat'] = 48.2 # lower right latitude
data['region']['lr']['lon'] = 11.9 # lower right longitude
data['region']['ul']['lat'] = 48.4 # upper left latitude
data['region']['ul']['lon'] = 11.6 # upper left longitude
data['region']['subset'] = 'yes'

data['single_file'] = 'yes'
```

(continues on next page)

(continued from previous page)

```
with open('test_config_file.yaml', 'wb') as stream:
    yaml.safe_dump(data, stream, default_flow_style=False,
                   explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Start pre-processing steps

[7]:
```
from sar_pre_processing.sar_pre_processor import *
import warnings
warnings.filterwarnings("ignore")

processing = SARPreProcessor(config='test_config_file.yaml')
processing.create_processing_file_list()
print('start step 1')
processing.pre_process_step1()
print('start step 2')
processing.pre_process_step2()
print('start step 3')
processing.pre_process_step3()
print('start add netcdf information')
processing.add_netcdf_information()
print('start create netcdf stack')
processing.create_netcdf_stack()
```

```
INFO:root:Found files within input folder: 1
INFO:root:year not specified
INFO:root:area of interest not specified
INFO:root:Number of found files that were double processed: 0.0
INFO:root:Number of found files with border issues: 0
INFO:root:area of interest specified
INFO:root:normalisation angle not specified, default value of 35 is used for processing
INFO:ComponentProgress:0
INFO:ComponentProgress:0
INFO:root:Process S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_027596_034B44_2DCF.
↪zip with SNAP.
```

```
start step 1
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: No valid orbit file found for 01-
↪JUL-2021 05:16:30.000000
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1B/2021/07
```

```
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
```

(continues on next page)

(continued from previous page)

```
OpenSearch: 38 total results on 1 pages.
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-06-01T00:00:000Z␣
↪TO 2021-06-31T24:00:000Z]
OpenSearch: 0 total results on 1 pages.
```

```
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: ApplyOrbit ignoring error and␣
↪continuing: java.io.IOException: No valid orbit file found for 01-JUL-2021 05:16:30.
↪000000
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1B/2021/07
```

```
...12%...24%..34%...46%
```

```
INFO: org.esa.snap.core.dataop.dem.ElevationFile: http retrieving http://step.esa.int/
↪auxdata/dem/SRTMGL1/N48E011.SRTMGL1.hgt.zip
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/dem/egm96/ww15mgh_b.zip
```

```
...58%..68%...80%... done.
```

```
INFO:root:0
INFO:root:Single image, no co-register of images necessary
INFO:root:multi temporal filter cannot applied to a single image, just single speckle␣
↪filter is applied
INFO:ComponentProgress:0
INFO:ComponentProgress:0
```

```
start step 2
start step 3
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...12%...24%.
```

```
27585 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
27602 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
27658 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
..36%...48%...60%...72%...84%.. done.
```

```
INFO:root:0
INFO:root:2021-08-29 17:54:26.666351
```

```
start add netcdf information
```

```
INFO:root:Number of scenes found for processing: 1
```

```
start create netcdf stack

Scene 1 of 1
/home/test/Desktop/data/step3/S1B_IW_SLC__1SDV_20210701T051738_20210701T051805_027596_
↪034B44_2DCF_GC_RC_No_Su_speckle.nc
```

4. View processed data

Load netcdf file with processed data

[8]:
```
import os
print(os.getcwd())
print(output_folder)
```

```
/home/test/Desktop/sar-pre-processing/docs/notebooks
/home/test/Desktop/data
```

[9]:
```
from netCDF4 import Dataset
import numpy as np

my_example_nc_file = os.path.join(output_folder, 'data.nc')
data = Dataset(my_example_nc_file, mode='r')
```

View information about dataset

[10]:
```
data
```

[10]:
```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
    dimensions(sizes): lat(1603), lon(2403), time(1)
    variables(dimensions): float32 time(time), float32 orbitdirection(time), float32␣
↪relorbit(time), float32 satellite(time), float32 lat(lat), float32 lon(lon), float32␣
↪theta(time, lat, lon), float32 sigma0_vv_single(time, lat, lon), float32 sigma0_vh_
↪single(time, lat, lon), float32 sigma0_vv_norm_single(time, lat, lon), float32 sigma0_
↪vh_norm_single(time, lat, lon)
    groups:
```

Read data from netcdf file

[11]:
```
data.variables['orbitdirection'][:]
data.variables['time'][:]
lons = data.variables['lon'][:]
lats = data.variables['lat'][:]
vv = data.variables['sigma0_vv_single'][:]

vv_units = data.variables['sigma0_vv_single'].units
```

Close netcdf file

[12]:
```
data.close()
```

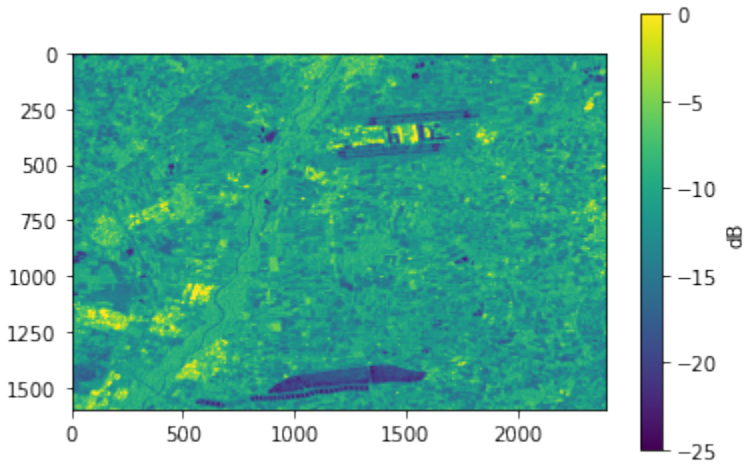
Plot vv polorized data

[13]:
```
%matplotlib inline
from ipywidgets import interactive
import matplotlib.pyplot as plt
import numpy as np

def f(x):
    # Problem: border pixel might be zero or negative
    # pixel eqal or smaller than zero are set to nan
    array = np.copy(vv[x])
    array[array <= 0] = np.nan
    # plot backscatter data in dB scale
    plt.imshow(10*np.log10(array))
    cbar = plt.colorbar()
    cbar.set_label('dB')
    plt.clim(-25, 0)

interactive_plot = interactive(f, x=(0,len(vv)-1))
interactive_plot
```

[13]:
```
interactive(children=(IntSlider(value=0, description='x', max=0), Output()), _dom_
↪classes=('widget-interact',)...
```



1.3.2 Example 2: Use default processing graphs to pre-process a time series of Sentinel-1 Level-1 SLC images

1. Requirements

- Installation of SenSARP
- Installation of ESA's SNAP Toolbox version >8.0.3
  - Currently only SNAP version 8.0 can be downloaded from the ESA website (https://step.esa.int/main/download/snap-download/). To update SNAP to a version >8.0.3 please start the SNAP software. You

will be asked if you want to search for update. After the updates are installed you need to restart SNAP to initialize the installed updates.

  - SNAP Toolbox need libgfortran for specific operations but currently libgfortran is not installed during the installation process of SNAP therefore you might use `sudo apt-get install gfortran`

- Sentinel-1 SLC data
  - Instruction how to download Sentinel 1 data are given in Section 2

2. Download sample data from Sentinel Data Hub

Option 1: Download data from Sentinel Data Hub manually or via python package sentinelsat

Create Account (https://scihub.copernicus.eu/dhus/#/self-registration). (**Attention: Problem by using Copernicus Open Access Hub might be that older data is offline and need to be triggered first**). More information can be found at https://scihub.copernicus.eu/userguide/DataRestoration. Instruction to manually download data from Copernicus Open Access Hub can be found at https://blogs.fu-berlin.de/reseda/esa-scihub/. You can also try to download the data via python package sentinelsat

How to use sentinelsat

[1]:
```
# connect to the API
from sentinelsat import SentinelAPI, read_geojson, geojson_to_wkt
from datetime import date
user = 'user'
password = 'password'
# initialize settings
api = SentinelAPI(user, password)
```

Search for available data

[2]:
```
# search by polygon (MNI test site coordinates), time, and SciHub query keywords
footprint = geojson_to_wkt(read_geojson('coordinates_mni.geojson'))
products = api.query(footprint,
                     date=('20210101', '20210110'),
                     platformname='Sentinel-1',
                     producttype='SLC')
print('Following products will be downloaded')
print(api.to_dataframe(products).title.values)

print('These {} product need {} Gb of disk space'.format(len(products), api.get_products_
↪size(products)))
```

```
Following products will be downloaded
['S1A_IW_SLC__1SDH_20210109T170737_20210109T170801_036064_043A0F_7B82'
 'S1A_IW_SLC__1SDV_20210108T051816_20210108T051844_036042_04393F_F99A'
 'S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_025044_02FB1E_E0E4'
 'S1A_IW_SLC__1SDV_20210104T165938_20210104T170006_035991_043767_329F'
 'S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_043767_A0D0'
 'S1B_IW_SLC__1SDV_20210103T170648_20210103T170715_024993_02F97C_C72A'
 'S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_02F8D2_6318'
```

(continues on next page)

(continued from previous page)

```
 'S1A_IW_SLC__1SDV_20210101T052626_20210101T052654_035940_0435A9_E0CB']
These 8 product need 59.76 Gb of disk space
```

Start download process (**Attention: might take a while and data will requries some free disk space**)

[3]:
```
# download all results from the search
# files will be downloaded to specified path
import os
path = os.path.expanduser('~/Desktop/data2')
try:
    os.makedirs(path)
except: FileExistsError
api.download_all(products, path)
```

```
Downloading products: 0product [00:00, ?product/s]
```

[3]:
```
ResultTuple(downloaded={'4c653f3d-2df7-4a76-ac3a-01abedbc0232': {'id': '4c653f3d-2df7-
↪4a76-ac3a-01abedbc0232', 'title': 'S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_
↪025044_02FB1E_E0E4', 'size': 4498240977, 'md5': '893B94C6E001FE655099F218950A5109',
↪'date': datetime.datetime(2021, 1, 7, 5, 25, 47, 737000), 'footprint': 'POLYGON((12.
↪658003 47.727371,9.287686 48.122635,9.664155 49.737373,13.147094 49.340591,12.658003␣
↪47.727371))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('4c653f3d-
↪2df7-4a76-ac3a-01abedbc0232')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 1, 7, 10, 45, 2, 244000), 'Ingestion Date': datetime.datetime(2021, 1,␣
↪7, 10, 38, 7, 194000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('4c653f3d-2df7-4a76-ac3a-01abedbc0232')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data2/S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_025044_
↪02FB1E_E0E4.zip'}, '5ade5549-a789-45c3-8272-c1dd61295340': {'id': '5ade5549-a789-45c3-
↪8272-c1dd61295340', 'title': 'S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_
↪043767_A0D0', 'size': 4439128243, 'md5': 'DFB54E48755274374CE427CB0341C866', 'date':␣
↪datetime.datetime(2021, 1, 4, 16, 59, 13, 974000), 'footprint': 'POLYGON((11.140376 48.
↪259541,14.598722 48.660851,14.967645 47.044518,11.617335 46.644329,11.140376 48.
↪259541))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('5ade5549-
↪a789-45c3-8272-c1dd61295340')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 1, 4, 22, 27, 41, 952000), 'Ingestion Date': datetime.datetime(2021, 1,␣
↪4, 22, 17, 8, 141000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('5ade5549-a789-45c3-8272-c1dd61295340')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data2/S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_
↪043767_A0D0.zip'}, '7dae304e-a8ad-4e8e-900e-fa98a79b89eb': {'id': '7dae304e-a8ad-4e8e-
↪900e-fa98a79b89eb', 'title': 'S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_
↪02F8D2_6318', 'size': 4399102981, 'md5': '52FF677638B5831E8C0FC0DB9D750CD3', 'date':␣
↪datetime.datetime(2021, 1, 2, 5, 17, 47, 426000), 'footprint': 'POLYGON((14.519586 46.
↪940491,11.161615 47.340321,11.533187 48.955601,15.000459 48.554497,14.519586 46.
↪940491))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('7dae304e-
↪a8ad-4e8e-900e-fa98a79b89eb')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 1, 2, 10, 47, 11, 479000), 'Ingestion Date': datetime.datetime(2021, 1,␣
↪2, 10, 37, 23, 997000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('7dae304e-a8ad-4e8e-900e-fa98a79b89eb')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data2/S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_
↪02F8D2_6318.zip'}, 'e31a15e5-8afd-4b30-b2fd-49f597afa11e': {'id': 'e31a15e5-8afd-4b30-
↪b2fd-49f597afa11e', 'title': 'S1A_IW_SLC__1SDV_20210104T165938_20210104T170006_035991_
↪043767_329F', 'size': 4659709650, 'md5': '46E3C2CE59AC05DB6E215820ACD82C0A', 'date':␣
↪datetime.datetime(2021, 1, 4, 16, 59, 38, 805000), 'footprint': 'POLYGON((10.656260 49.
↪801006,14.231164 50.204681,14.622996 48.531536,11.169555 48.129627,10.656260 49.
↪801006))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('e31a15e5-
↪8afd-4b30-b2fd-49f597afa11e')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 1, 4, 22, 28, 3, 971000), 'Ingestion Date': datetime.datetime(2021, 1,␣
↪4, 22, 19, 22, 84000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('e31a15e5-8afd-4b30-b2fd-49f597afa11e')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data2/S1A_IW_SLC__1SDV_20210104T165938_20210104T170006_035991_
```

(continues on next page)

(continued from previous page)

Option 2: Manually search and download data from Alaska Satellite Facility (ASF)

You can search for Sentinel-1 data at https://search.asf.alaska.edu/. A NASA EOSDIS Earthdata Login account is required for downloading data and tools from ASF. Registering for an Earthdata Login account is free (https://urs.earthdata.nasa.gov/home). Instructions how to download data from ASF can be found at https://asf.alaska.edu/wp-content/uploads/2019/02/asf_datarecipe_bulk_download_from_vertex_python_script_v1.pdf.

3. Use default processing graph to pre-process a time series of Sentinel-1 Level-1 SLC images

Set paths for

- input_folder (path to stored Sentinel-1 SLC data (zip files) e.g. "~/Downloads")
- output_folder (path where processed data will be stored e.g. "~/output")
- gpt_loction (gpt is located in the bin folder of your SNAP installation)'

[4]:
```
input_folder = path
output_folder = path
gpt_location = os.path.expanduser('~/snap/bin/gpt')
```

Create config file with information about input, output and gpt location

[5]:
```
import yaml

with open('sample_config_file.yaml') as stream:
   data = yaml.safe_load(stream)

data['input_folder'] = input_folder
data['output_folder'] = output_folder
data['gpt'] = gpt_location

with open('test_config_file.yaml', 'wb') as stream:
   yaml.safe_dump(data, stream, default_flow_style=False,
                  explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Set optional config options

[6]:
```
with open('test_config_file.yaml') as stream:
   data = yaml.safe_load(stream)

# Filter option
## Filter via year of interes
data['year'] = '2021'

## Define region of interest
data['region']['lr']['lat'] = 48.2 # lower right latitude
data['region']['lr']['lon'] = 11.9 # lower right longitude
data['region']['ul']['lat'] = 48.4 # upper left latitude
data['region']['ul']['lon'] = 11.6 # upper left longitude
```

(continues on next page)

(continued from previous page)

```
data['region']['subset'] = 'yes'

## Define multi-temporal filtering properties
data['speckle_filter']['multi_temporal']['apply'] = 'yes'
data['speckle_filter']['multi_temporal']['files'] = '5' # Number of files used for multi␣
↪temporal filtering

## Define incidence angle for normalization
data['normalization_angle'] = '35'

with open('test_config_file.yaml', 'wb') as stream:
   yaml.safe_dump(data, stream, default_flow_style=False,
                  explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Start pre-processing steps

[7]:
```
from sar_pre_processing.sar_pre_processor import *
import warnings
warnings.filterwarnings("ignore")

processing = SARPreProcessor(config='test_config_file.yaml')
processing.create_processing_file_list()
print('start step 1')
processing.pre_process_step1()
print('start step 2')
processing.pre_process_step2()
print('start step 3')
processing.pre_process_step3()
print('start add netcdf information')
processing.add_netcdf_information()
print('start create netcdf stack')
processing.create_netcdf_stack()
```

```
INFO:root:Found files within input folder: 10
INFO:root:Number of found files for year 2021: 10
INFO:root:area of interest not specified
INFO:root:Number of found files that were double processed: 0.0
INFO:root:Number of found files with border issues: 4
INFO:root:area of interest specified
INFO:root:normalisation angle not specified, default value of 35 is used for processing
INFO:ComponentProgress:0
INFO:ComponentProgress:0
INFO:root:Process S1A_IW_SLC__1SDH_20210109T170737_20210109T170801_036064_043A0F_7B82.
↪zip with SNAP.
```

```
start step 1
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1A/2021/01/S1A_OPER_AUX_POEORB_
↪OPOD_20210129T120952_V20210108T225942_20210110T005942.EOF.zip

INFO:root:1
INFO:ComponentProgress:10
INFO:ComponentProgress:10
INFO:root:Process S1A_IW_SLC__1SDV_20210101T052626_20210101T052654_035940_0435A9_E0CB.
↪zip with SNAP.
```

```
 done.
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1A/2020/12/S1A_OPER_AUX_POEORB_
↪OPOD_20210121T121629_V20201231T225942_20210102T005942.EOF.zip
```

```
...12%...24%...36%...48%...60%...72%...84%.. done.
```

```
INFO:root:0
INFO:ComponentProgress:20
INFO:ComponentProgress:20
INFO:root:Process S1A_IW_SLC__1SDV_20210108T051816_20210108T051844_036042_04393F_F99A.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1A/2021/01/S1A_OPER_AUX_POEORB_
↪OPOD_20210128T121545_V20210107T225942_20210109T005942.EOF.zip
```

```
...11%...23%...34%...46%...57%...69%...80%... done.
```

```
INFO:root:0
INFO:ComponentProgress:30
INFO:ComponentProgress:30
INFO:root:Process S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_02F8D2_6318.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1B/2021/01/S1B_OPER_AUX_POEORB_
↪OPOD_20210122T111559_V20210101T225942_20210103T005942.EOF.zip
```

```
...12%...24%..34%...46%...58%..68%...80%... done.
```

```
INFO:root:0
INFO:ComponentProgress:40
INFO:ComponentProgress:40
INFO:root:Process S1B_IW_SLC__1SDV_20210103T170648_20210103T170715_024993_02F97C_C72A.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1B/2021/01/S1B_OPER_AUX_POEORB_
↪OPOD_20210123T111611_V20210102T225942_20210104T005942.EOF.zip
```

```
..10%..20%..31%..41%...51%..62%..72%..83%.. done.
```

```
INFO:root:0
INFO:ComponentProgress:50
INFO:ComponentProgress:50
INFO:root:Process S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_025044_02FB1E_E0E4.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1B/2021/01/S1B_OPER_AUX_POEORB_
↪OPOD_20210127T111615_V20210106T225942_20210108T005942.EOF.zip
```

```
...12%...24%..34%...46%...58%..68%...80%... done.
```

```
INFO:root:0
INFO:ComponentProgress:60
INFO:ComponentProgress:60
INFO:root:Process S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_043767_A0D0.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1A/2021/01/S1A_OPER_AUX_POEORB_
↪OPOD_20210124T121622_V20210103T225942_20210105T005942.EOF.zip
```

```
...12%...24%...34%...46%...58%...68%...80%...90% done.
```

```
INFO:root:0
INFO:ComponentProgress:70
INFO:ComponentProgress:70
INFO:root:Process S1B_IW_SLC__1SDV_20210110T165830_20210110T165857_025095_02FCB9_7015.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.engine_utilities.download.DownloadableContentImpl: http retrieving␣
↪http://step.esa.int/auxdata/orbits/Sentinel-1/POEORB/S1B/2021/01/S1B_OPER_AUX_POEORB_
↪OPOD_20210130T110934_V20210109T225942_20210111T005942.EOF.zip
```

```
...11%...23%...34%...45%..55%...66%...78%...90% done.
```

```
INFO:root:0
INFO:root:skip processing for /home/test/Desktop/data2/S1A_IW_SLC__1SDH_20210109T170737_
↪20210109T170801_036064_043A0F_7B82.zip. File does not exist
INFO:root:skip processing for /home/test/Desktop/data2/S1A_IW_SLC__1SDV_20210104T165938_
↪20210104T170006_035991_043767_329F.zip. File does not exist
INFO:root:skip processing for /home/test/Desktop/data2/S1B_IW_SLC__1SDV_20210110T165855_
↪20210110T165922_025095_02FCB9_4BB8.zip. File does not exist
INFO:ComponentProgress:0
INFO:ComponentProgress:0
INFO:root:Scene 1 of 7
INFO:root:Process S1A_IW_SLC__1SDV_20210101T052626_20210101T052654_035940_0435A9_E0CB_GC_
↪RC_No_Su.dim with SNAP.
```

```
start step 2
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas7850174183669672141/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas7850174183669672141/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas7850174183669672141/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas7850174183669672141/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas7850174183669672141
INFO:root:0
INFO:root:2021-08-29 18:57:58.801427
INFO:ComponentProgress:14
INFO:ComponentProgress:14
INFO:root:Scene 2 of 7
INFO:root:Process S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_043767_A0D0_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
WARNING: org.jlinda.core.Baseline: Max. error bperp modeling at 3D datapoints: 951621.
↪6928487846m
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas8286044915352018441/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas8286044915352018441/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas8286044915352018441/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas8286044915352018441/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas8286044915352018441
INFO:root:0
INFO:root:2021-08-29 18:59:12.674310
INFO:ComponentProgress:28
INFO:ComponentProgress:28
INFO:root:Scene 3 of 7
INFO:root:Process S1A_IW_SLC__1SDV_20210108T051816_20210108T051844_036042_04393F_F99A_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas1941578135814371655/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas1941578135814371655/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas1941578135814371655/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas1941578135814371655/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas1941578135814371655
INFO:root:0
INFO:root:2021-08-29 19:00:26.270922
INFO:ComponentProgress:42
INFO:ComponentProgress:42
INFO:root:Scene 4 of 7
INFO:root:Process S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_02F8D2_6318_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas5133865250962490398/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas5133865250962490398/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas5133865250962490398/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas5133865250962490398/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas5133865250962490398
INFO:root:0
INFO:root:2021-08-29 19:01:57.069051
INFO:ComponentProgress:57
INFO:ComponentProgress:57
INFO:root:Scene 5 of 7
INFO:root:Process S1B_IW_SLC__1SDV_20210103T170648_20210103T170715_024993_02F97C_C72A_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas8785853447239672772/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas8785853447239672772/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas8785853447239672772/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas8785853447239672772/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas8785853447239672772
INFO:root:0
INFO:root:2021-08-29 19:03:24.549310
INFO:ComponentProgress:71
INFO:ComponentProgress:71
INFO:root:Scene 6 of 7
INFO:root:Process S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_025044_02FB1E_E0E4_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas7411179791605111946/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas7411179791605111946/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas7411179791605111946/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas7411179791605111946/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas7411179791605111946
INFO:root:0
INFO:root:2021-08-29 19:04:35.352154
INFO:ComponentProgress:85
INFO:ComponentProgress:85
INFO:root:Scene 7 of 7
INFO:root:Process S1B_IW_SLC__1SDV_20210110T165830_20210110T165857_025095_02FCB9_7015_GC_
↪RC_No_Su.dim with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
INFO: org.esa.snap.core.datamodel.Product: raster width 2403 not equal to 2404
WARNING: org.jlinda.core.Baseline: Max. error bperp modeling at 3D datapoints: 1002303.
↪1955448636m
```

```
...12%...25%..35%..45%...57%..67%..77%...89% done.
```

```
-- org.jblas INFO Deleting /tmp/jblas2590257929860331368/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas2590257929860331368/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas2590257929860331368/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas2590257929860331368/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas2590257929860331368
INFO:root:0
INFO:root:2021-08-29 19:05:54.247359
```

(continues on next page)

(continued from previous page)

```
INFO:root:skip processing for /home/test/Desktop/data2/S1A_IW_SLC__1SDH_20210109T170737_
↪20210109T170801_036064_043A0F_7B82.zip. File /home/test/Desktop/data2/step2/S1A_IW_SLC_
↪_1SDH_20210109T170737_20210109T170801_036064_043A0F_7B82_GC_RC_No_Su_Co.dim does not␣
↪exist.
INFO:root:skip processing for /home/test/Desktop/data2/S1A_IW_SLC__1SDV_20210104T165938_
↪20210104T170006_035991_043767_329F.zip. File /home/test/Desktop/data2/step2/S1A_IW_SLC_
↪_1SDV_20210104T165938_20210104T170006_035991_043767_329F_GC_RC_No_Su_Co.dim does not␣
↪exist.
INFO:root:skip processing for /home/test/Desktop/data2/S1B_IW_SLC__1SDV_20210110T165855_
↪20210110T165922_025095_02FCB9_4BB8.zip. File /home/test/Desktop/data2/step2/S1B_IW_SLC_
↪_1SDV_20210110T165855_20210110T165922_025095_02FCB9_4BB8_GC_RC_No_Su_Co.dim does not␣
↪exist.
INFO:ComponentProgress:0
INFO:ComponentProgress:0
```

```
start step 3
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...10%...22%...34%...45%...
```

```
29455 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
29500 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
29501 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
55%...67%...79%...90% done.
```

```
INFO:root:0
INFO:root:2021-08-29 19:08:56.411130
INFO:ComponentProgress:14
INFO:ComponentProgress:14
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...10%...22%...34%...45%...
```

```
21992 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
22011 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
22012 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
55%...67%...79%...90% done.
```

```
INFO:root:0
INFO:root:2021-08-29 19:11:42.027038
INFO:ComponentProgress:28
INFO:ComponentProgress:28
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...10%...22%...34%...45%...
```

```
21802 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
21810 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
21828 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
55%...67%...79%...90%
```

```
INFO:root:0
INFO:root:2021-08-29 19:14:12.068346
INFO:ComponentProgress:42
INFO:ComponentProgress:42
```

```
 done.
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...10%...22%...34%...45%...
```

```
13418 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
13434 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
13435 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
55%...67%...79%...90% done.

INFO:root:0
INFO:root:2021-08-29 19:16:33.320428
INFO:ComponentProgress:57
INFO:ComponentProgress:57
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.

Executing processing graph

INFO: org.hsqldb.persist.Logger: dataFileCache open start

...10%...22%...34%...45%...

22425 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
22442 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
22448 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0

55%...67%...79%...90% done.

INFO:root:0
INFO:root:2021-08-29 19:19:14.782446
INFO:ComponentProgress:71
INFO:ComponentProgress:71
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.

Executing processing graph

INFO: org.hsqldb.persist.Logger: dataFileCache open start

...10%...22%...34%...45%...

22348 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
22375 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
22387 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0

55%...67%...79%...90% done.

INFO:root:0
INFO:root:2021-08-29 19:21:56.809279
INFO:ComponentProgress:85
INFO:ComponentProgress:85
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
```

(continues on next page)

(continued from previous page)

```
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: GDAL 3.0.4 found on system. JNI driver will␣
↪be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Installed GDAL 3.0.4 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...10%...22%...34%...45%...
```

```
21504 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
21543 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
21543 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
55%...67%...79%...90% done.
```

```
INFO:root:0
INFO:root:2021-08-29 19:24:31.369057
```

```
start add netcdf information
```

```
INFO:root:Number of scenes found for processing: 7
```

```
start create netcdf stack

Scene 1 of 7
/home/test/Desktop/data2/step3/S1A_IW_SLC__1SDV_20210101T052626_20210101T052654_035940_
↪0435A9_E0CB_GC_RC_No_Su_Co_speckle.nc

Scene 2 of 7
/home/test/Desktop/data2/step3/S1B_IW_SLC__1SDV_20210102T051747_20210102T051814_024971_
↪02F8D2_6318_GC_RC_No_Su_Co_speckle.nc

Scene 3 of 7
/home/test/Desktop/data2/step3/S1B_IW_SLC__1SDV_20210103T170648_20210103T170715_024993_
↪02F97C_C72A_GC_RC_No_Su_Co_speckle.nc

Scene 4 of 7
/home/test/Desktop/data2/step3/S1A_IW_SLC__1SDV_20210104T165913_20210104T165940_035991_
↪043767_A0D0_GC_RC_No_Su_Co_speckle.nc

Scene 5 of 7
/home/test/Desktop/data2/step3/S1B_IW_SLC__1SDV_20210107T052547_20210107T052614_025044_
↪02FB1E_E0E4_GC_RC_No_Su_Co_speckle.nc

Scene 6 of 7
/home/test/Desktop/data2/step3/S1A_IW_SLC__1SDV_20210108T051816_20210108T051844_036042_
↪04393F_F99A_GC_RC_No_Su_Co_speckle.nc

Scene 7 of 7
/home/test/Desktop/data2/step3/S1B_IW_SLC__1SDV_20210110T165830_20210110T165857_025095_
↪02FCB9_7015_GC_RC_No_Su_Co_speckle.nc
```

4. View processed data

Load netcdf file with processed data

[8]:
```
import os
print(os.getcwd())
print(output_folder)
```

```
/home/test/Desktop/sar-pre-processing/docs/notebooks
/home/test/Desktop/data2
```

[9]:
```
from netCDF4 import Dataset
import numpy as np

my_example_nc_file = os.path.join(output_folder, 'data2.nc')
data = Dataset(my_example_nc_file, mode='r')
```

View information about dataset

[10]:
```
data
```

[10]:
```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
    dimensions(sizes): lat(1603), lon(2403), time(7)
    variables(dimensions): float32 time(time), float32 orbitdirection(time), float32␣
↪relorbit(time), float32 satellite(time), float32 lat(lat), float32 lon(lon), float32␣
↪theta(time, lat, lon), float32 sigma0_vv_norm_multi(time, lat, lon), float32 sigma0_vh_
↪norm_multi(time, lat, lon), float32 sigma0_vv_multi(time, lat, lon), float32 sigma0_vh_
↪multi(time, lat, lon), float32 sigma0_vv_single(time, lat, lon), float32 sigma0_vh_
↪single(time, lat, lon), float32 sigma0_vv_norm_single(time, lat, lon), float32 sigma0_
↪vh_norm_single(time, lat, lon)
    groups:
```

Read data from netcdf file

[11]:
```
data.variables['orbitdirection'][:]
data.variables['time'][:]
lons = data.variables['lon'][:]
lats = data.variables['lat'][:]
vv = data.variables['sigma0_vv_single'][:]

vv_units = data.variables['sigma0_vv_single'].units
```

Close netcdf file

[12]:
```
data.close()
```

Plot vv polorized data

[13]:
```
%matplotlib inline
from ipywidgets import interactive
import matplotlib.pyplot as plt
import numpy as np

def f(x):
```

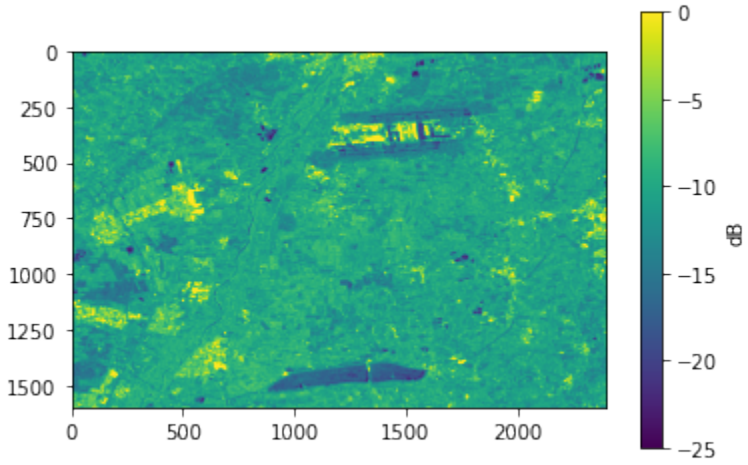
(continues on next page)

(continued from previous page)

```
    # Problem: border pixel might be zero or negative
    # pixel eqal or smaller than zero are set to nan
    array = np.copy(vv[x])
    array[array <= 0] = np.nan
    # plot backscatter data in dB scale
    plt.imshow(10*np.log10(array))
    cbar = plt.colorbar()
    cbar.set_label('dB')
    plt.clim(-25, 0)

interactive_plot = interactive(f, x=(0,len(vv)-1))
interactive_plot
```

[13]:
```
interactive(children=(IntSlider(value=3, description='x', max=6), Output()), _dom_
↪classes=('widget-interact',)...
```



[ ]:

1.3.3 Example 3: Use user defined processing graphs to pre-process a time series of Sentinel-1 images

1. Requirements

- Installation of SenSARP
- Installation of ESA's SNAP Toolbox version >8.0.3
  - Currently only SNAP version 8.0 can be downloaded from the ESA website (https://step.esa.int/main/download/snap-download/). To update SNAP to a version >8.0.3 please start the SNAP software. You will be asked if you want to search for update. After the updates are installed you need to restart SNAP to initialize the installed updates.
  - SNAP Toolbox need libgfortran for specific operations but currently libgfortran is not installed during the installation process of SNAP therefore you might use `sudo apt-get install gfortran`
- Sentinel-1 SLC data
  - Instruction how to download Sentinel 1 data are given in Section 2

2. Download sample data from Sentinel Data Hub

Option 1: Download data from Sentinel Data Hub manually or via python package sentinelsat

Create Account (https://scihub.copernicus.eu/dhus/#/self-registration). (**Attention: Problem by using Copernicus Open Access Hub might be that older data is offline and need to be triggered first**). More information can be found at https://scihub.copernicus.eu/userguide/DataRestoration. Instruction to manually download data from Copernicus Open Access Hub can be found at https://blogs.fu-berlin.de/reseda/esa-scihub/. You can also try to download the data via python package sentinelsat

[1]:
```
# connect to the API
from sentinelsat import SentinelAPI, read_geojson, geojson_to_wkt
from datetime import date
user = 'user'
password = 'password'
# initialize settings
api = SentinelAPI(user, password)
```

Search for available data

[2]:
```
# search by polygon (MNI test site coordinates), time, and SciHub query keywords
footprint = geojson_to_wkt(read_geojson('coordinates_mni.geojson'))
products = api.query(footprint,
                     date=('20210704', '20210708'),
                     platformname='Sentinel-1',
                     producttype='GRD')
print('Following products will be downloaded')
print(api.to_dataframe(products).title.values)

print('These {} product need {} Gb of disk space'.format(len(products), api.get_products_
↪size(products)))
```

```
INFO:sentinelsat.SentinelAPI:Found 2 products
```

```
Following products will be downloaded
['S1A_IW_GRDH_1SDV_20210707T051822_20210707T051847_038667_04901E_1B6B'
 'S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_034D60_888E']
These 2 product need 3.28 Gb of disk space
```

Start download process (**Attention: might take a while and data will requries some free disk space**)

[3]:
```
# download all results from the search
# files will be downloaded to specified path
import os
path = os.path.expanduser('~/Desktop/data4')
try:
    os.makedirs(path)
except: FileExistsError
api.download_all(products, path)
```

```
INFO:sentinelsat.SentinelAPI:Will download 2 products using 4 workers
```

```
Downloading products:   0%|          | 0/2 [00:00<?, ?product/s]
```

```
Downloading S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_034D60_888E.zip:   0
↪%|          | 0.00/986...
```

```
Downloading S1A_IW_GRDH_1SDV_20210707T051822_20210707T051847_038667_04901E_1B6B.zip:   0
↪%|          | 0.00/983...
```

```
MD5 checksumming:   0%|          | 0.00/986M [00:00<?, ?B/s]
```

```
MD5 checksumming:   0%|          | 0.00/983M [00:00<?, ?B/s]
```

[3]:
```
ResultTuple(downloaded={'ce796d52-4f70-48f0-87bf-d698ec372e9f': {'id': 'ce796d52-4f70-
↪48f0-87bf-d698ec372e9f', 'title': 'S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_
↪027669_034D60_888E', 'size': 986204913, 'md5': '56c6d22b25610fb30b01212d0b35101b',
↪'date': datetime.datetime(2021, 7, 6, 5, 25, 51, 753000), 'footprint': 'POLYGON((12.
↪682894 47.832607,9.216668 48.236717,9.567789 49.733212,13.138339 49.327972,12.682894␣
↪47.832607))', 'url': "https://apihub.copernicus.eu/apihub/odata/v1/Products('ce796d52-
↪4f70-48f0-87bf-d698ec372e9f')/$value", 'Online': True, 'Creation Date': datetime.
↪datetime(2021, 7, 6, 7, 26, 2, 888000), 'Ingestion Date': datetime.datetime(2021, 7, 6,
↪ 7, 24, 14, 101000), 'quicklook_url': "https://apihub.copernicus.eu/apihub/odata/v1/
↪Products('ce796d52-4f70-48f0-87bf-d698ec372e9f')/Products('Quicklook')/$value", 'path':
↪ '/home/test/Desktop/data4/S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_
↪034D60_888E.zip', 'downloaded_bytes': 986204913}, 'caf75225-cb85-4410-81cd-ff6ef52256ed
↪': {'id': 'caf75225-cb85-4410-81cd-ff6ef52256ed', 'title': 'S1A_IW_GRDH_1SDV_
↪20210707T051822_20210707T051847_038667_04901E_1B6B', 'size': 983271059, 'md5':
↪'4ba73f80273607ea8106cce310ba4d93', 'date': datetime.datetime(2021, 7, 7, 5, 18, 22,␣
↪410000), 'footprint': 'POLYGON((14.717325 47.687527,11.218726 48.096413,11.575086 49.
↪592331,15.178372 49.182144,14.717325 47.687527))', 'url': "https://apihub.copernicus.
↪eu/apihub/odata/v1/Products('caf75225-cb85-4410-81cd-ff6ef52256ed')/$value", 'Online':␣
↪True, 'Creation Date': datetime.datetime(2021, 7, 7, 10, 32, 2, 46000), 'Ingestion Date
↪': datetime.datetime(2021, 7, 7, 10, 30, 11, 282000), 'quicklook_url': "https://apihub.
↪copernicus.eu/apihub/odata/v1/Products('caf75225-cb85-4410-81cd-ff6ef52256ed')/
↪Products('Quicklook')/$value", 'path': '/home/test/Desktop/data4/S1A_IW_GRDH_1SDV_
↪20210707T051822_20210707T051847_038667_04901E_1B6B.zip', 'downloaded_bytes': 983271059}
↪}, retrieval_triggered={}, failed={})
```

Option 2: Download data from NASA Earth Data Search

You can search for Sentinel-1 data at https://search.asf.alaska.edu/. A NASA EOSDIS Earthdata Login account is required for downloading data and tools from ASF. Registering for an Earthdata Login account is free (https://urs.earthdata.nasa.gov/home). Instructions how to download data from ASF can be found at https://asf.alaska.edu/wp-content/uploads/2019/02/asf_datarecipe_bulk_download_from_vertex_python_script_v1.pdf.

3. Use user defined xml graph to process Sentinel-1 data (test case with Sentinel-1 GRD product)

Set paths for

- input_folder (path to stored Sentinel-1 SLC data (zip files) e.g. "~/Downloads")
- output_folder (path where processed data will be stored e.g. "~/output")
- gpt_loction (gpt is located in the bin folder of your SNAP installation)'

[5]:
```
input_folder = path
output_folder = path
gpt_location = os.path.expanduser('~/snap/bin/gpt')
```

Create config file with information about input, output and gpt location

[6]:
```
import yaml

with open('sample_config_file.yaml') as stream:
   data = yaml.safe_load(stream)

data['input_folder'] = input_folder
data['output_folder'] = output_folder
data['gpt'] = gpt_location

with open('test_config_file.yaml', 'wb') as stream:
   yaml.safe_dump(data, stream, default_flow_style=False,
                  explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Expert user might create their own processing chain and use the functionally of SenSARP to automate the processing on scale

1. Create a user defined xml graph via SNAP's Graph Builder
   - information how to use SNAP's Graph Builder can be found at the step forum (https://forum.step.esa.int/t/graph-builder/5403)
2. Modify user defined xml graph to be used by SenSARP
   - Needed modification
     - Placeholder for input image within xml graph need to be specified
     - Placeholder for output image within xml graph need to be specified
     - If subset operator is used within user defined xml graph a placeholder for subset extent need to be specified

Additional functionality of SenSARP that can be used with user defined xml graph

- Filter option (year, area of interest)
- default pre_process_step2 (co-registration of time series images)
- creation of netCDF4 stack containing all processed images and information about orbit direction, relative orbit, satellite name

Find and show xml parts that need to be changed

[7]:
```
# Load xml graph
import xml.etree.ElementTree as ETree
user_xml_graph = '/home/test/Desktop/sar-pre-processing/sar_pre_processing/user_defined_
↪graphs/example_of_expert_user.xml'
tree = ETree.parse(user_xml_graph)
root = tree.getroot()

# find Operators Read, Write and Subset
read = root.find("./node/[operator='Read']")
read2 = read.find("./parameters[@class='com.bc.ceres.binding.dom.XppDomElement']")
write = root.find("./node/[operator='Write']")
subset = root.find("./node/[operator='Subset']")

# Old Version of user defined xml graph
print(ETree.tostring(read, encoding='utf8').decode('utf8'))
```

(continues on next page)

(continued from previous page)

```
print(ETree.tostring(write, encoding='utf8').decode('utf8'))
print(ETree.tostring(subset, encoding='utf8').decode('utf8'))
```

```
<?xml version='1.0' encoding='utf8'?>
<node id="Read">
    <operator>Read</operator>
    <sources />
    <parameters class="com.bc.ceres.binding.dom.XppDomElement" />
  </node>

<?xml version='1.0' encoding='utf8'?>
<node id="Write">
    <operator>Write</operator>
    <sources>
      <sourceProduct refid="Subset" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <file>/home/wodan/target.dim</file>
      <formatName>BEAM-DIMAP</formatName>
    </parameters>
  </node>

<?xml version='1.0' encoding='utf8'?>
<node id="Subset">
    <operator>Subset</operator>
    <sources>
      <sourceProduct refid="Terrain-Correction" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <sourceBands />
      <region>0,0,0,0</region>
      <referenceBand />
      <geoRegion />
      <subSamplingX>1</subSamplingX>
      <subSamplingY>1</subSamplingY>
      <fullSwath>false</fullSwath>
      <tiePointGridNames />
      <copyMetadata>true</copyMetadata>
    </parameters>
  </node>
```

Modify xml parts so the xml-graph can be used by SenSARP

[8]:
```
#Apply changes to user defined xml graph
ETree.SubElement(read2, "file").text = "$input"
for output in write.iter('file'):
    output_new = '$output'
    output.text = output_new
for area in subset.iter('geoRegion'):
    area_new = '$area'
    area.text = area_new
```

(continues on next page)

(continued from previous page)

```
tree.write('/home/test/Desktop/sar-pre-processing/sar_pre_processing/user_defined_graphs/
↪example_of_expert_user_modi.xml')

# print modified version of operators Read, Write and Subset
print(ETree.tostring(read, encoding='utf8').decode('utf8'))
print(ETree.tostring(write, encoding='utf8').decode('utf8'))
print(ETree.tostring(subset, encoding='utf8').decode('utf8'))
```

```
<?xml version='1.0' encoding='utf8'?>
<node id="Read">
    <operator>Read</operator>
    <sources />
    <parameters class="com.bc.ceres.binding.dom.XppDomElement"><file>$input</file></
↪parameters>
  </node>

<?xml version='1.0' encoding='utf8'?>
<node id="Write">
    <operator>Write</operator>
    <sources>
      <sourceProduct refid="Subset" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <file>$output</file>
      <formatName>BEAM-DIMAP</formatName>
    </parameters>
  </node>

<?xml version='1.0' encoding='utf8'?>
<node id="Subset">
    <operator>Subset</operator>
    <sources>
      <sourceProduct refid="Terrain-Correction" />
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
      <sourceBands />
      <region>0,0,0,0</region>
      <referenceBand />
      <geoRegion>$area</geoRegion>
      <subSamplingX>1</subSamplingX>
      <subSamplingY>1</subSamplingY>
      <fullSwath>false</fullSwath>
      <tiePointGridNames />
      <copyMetadata>true</copyMetadata>
    </parameters>
  </node>
```

Set additional config options

[9]:
```
with open('test_config_file.yaml') as stream:
   data = yaml.safe_load(stream)
```

(continues on next page)

(continued from previous page)

```
# Filter option
## Filter via year of interest
data['year'] = '2021'

## Define region of interest
data['region']['lr']['lat'] = 48.2 # lower right latitude
data['region']['lr']['lon'] = 11.9 # lower right longitude
data['region']['ul']['lat'] = 48.4 # upper left latitude
data['region']['ul']['lon'] = 11.6 # upper left longitude
data['region']['subset'] = 'yes'

## Set options to use user defined xml graph
data['use_user_defined_graphs'] = 'yes'
data['xml_graph_path'] = '/home/test/Desktop/sar-pre-processing/sar_pre_processing/user_
↪defined_graphs'
data['pre_process_step1'] = 'example_of_expert_user_modi.xml'

with open('test_config_file.yaml', 'wb') as stream:
   yaml.safe_dump(data, stream, default_flow_style=False,
                  explicit_start=True, allow_unicode=True, encoding='utf-8')
```

Start pre-processing steps

[10]:
```
from sar_pre_processing.sar_pre_processor import *
import warnings
warnings.filterwarnings("ignore")

processing = SARPreProcessor(config='test_config_file.yaml')
processing.create_processing_file_list()
print('start step 1')
processing.pre_process_step1()
print('start step 2')
processing.pre_process_step2()
print('start step 3')
processing.pre_process_step3()
print('start add netcdf information')
processing.add_netcdf_information()
print('start create netcdf stack')
processing.create_netcdf_stack()
```

```
INFO:root:Found files within input folder: 2
INFO:root:Number of found files for year 2021: 2
INFO:root:area of interest not specified
INFO:root:Number of found files that were double processed: 0.0
INFO:root:Number of found files with border issues: 0
INFO:root:area of interest specified
INFO:root:normalisation angle not specified, default value of 35 is used for processing
INFO:ComponentProgress:0
INFO:ComponentProgress:0
INFO:root:Process S1A_IW_GRDH_1SDV_20210707T051822_20210707T051847_038667_04901E_1B6B.
↪zip with SNAP.
```

```
start step 1
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Incompatible GDAL 3.3.1 found on system.␣
↪Internal GDAL 3.0.0 from distribution will be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: No valid orbit file found for 07-
↪JUL-2021 05:17:13.000000
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1A/2021/07
```

```
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:A AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
OpenSearch: 35 total results on 1 pages.
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:A AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:A AND producttype:AUX_RESORB AND beginposition:[2021-06-01T00:00:000Z␣
↪TO 2021-06-31T24:00:000Z]
OpenSearch: 0 total results on 1 pages.
```

```
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: ApplyOrbit ignoring error and␣
↪continuing: java.io.IOException: No valid orbit file found for 07-JUL-2021 05:17:13.
↪000000
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1A/2021/07
```

```
version = 3.31
...10%...21%...33%...44%...54%...65%...77%...88%. done.
```

```
INFO:root:0
INFO:ComponentProgress:50
INFO:ComponentProgress:50
INFO:root:Process S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_034D60_888E.
↪zip with SNAP.
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Incompatible GDAL 3.3.1 found on system.␣
↪Internal GDAL 3.0.0 from distribution will be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: No valid orbit file found for 06-
↪JUL-2021 05:24:42.000000
```

(continues on next page)

(continued from previous page)

```
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1B/2021/07
```

```
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
OpenSearch: 36 total results on 1 pages.
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-07-01T00:00:000Z␣
↪TO 2021-07-31T24:00:000Z]
OpenSearch: https://scihub.copernicus.eu/gnss/search?q=platformname:Sentinel-1 AND␣
↪platformnumber:B AND producttype:AUX_RESORB AND beginposition:[2021-06-01T00:00:000Z␣
↪TO 2021-06-31T24:00:000Z]
OpenSearch: 0 total results on 1 pages.
```

```
WARNING: org.esa.s1tbx.sar.gpf.orbits.ApplyOrbitFileOp: ApplyOrbit ignoring error and␣
↪continuing: java.io.IOException: No valid orbit file found for 06-JUL-2021 05:24:42.
↪000000
Orbit files may be downloaded from https://scihub.copernicus.eu/gnss/odata/v1/
and placed in /home/test/.snap/auxdata/Orbits/Sentinel-1/POEORB/S1B/2021/07
```

```
version = 3.31
...11%...23%...34%...46%...57%...69%...80%... done.
```

```
INFO:root:0
INFO:ComponentProgress:0
INFO:ComponentProgress:0
INFO:root:Scene 1 of 2
INFO:root:Process S1A_IW_GRDH_1SDV_20210707T051822_20210707T051847_038667_04901E_1B6B_GC_
↪RC_No_Su.dim with SNAP.
```

```
start step 2
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Incompatible GDAL 3.3.1 found on system.␣
↪Internal GDAL 3.0.0 from distribution will be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
```

```
...
```

```
14638 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
14676 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
14676 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
11%...22%...32%...43%...53%...64%...75%...85%..
```

```
-- org.jblas INFO Deleting /tmp/jblas4819000381521070295/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas4819000381521070295/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas4819000381521070295/libjblas_arch_flavor.so
```

(continues on next page)

(continued from previous page)

```
-- org.jblas INFO Deleting /tmp/jblas4819000381521070295/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas4819000381521070295
INFO:root:0
INFO:root:2021-08-29 20:28:51.715338
INFO:ComponentProgress:50
INFO:ComponentProgress:50
INFO:root:Scene 2 of 2
INFO:root:Process S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_034D60_888E_GC_
↪RC_No_Su.dim with SNAP.
```

```
 done.
```

```
INFO: org.esa.snap.core.gpf.operators.tooladapter.ToolAdapterIO: Initializing external␣
↪tool adapters
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Incompatible GDAL 3.3.1 found on system.␣
↪Internal GDAL 3.0.0 from distribution will be used.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
INFO: org.esa.snap.core.util.EngineVersionCheckActivator: Please check regularly for new␣
↪updates for the best SNAP experience.
INFO: org.esa.s2tbx.dataio.gdal.GDALVersion: Internal GDAL 3.0.0 set to be used by SNAP.
```

```
Executing processing graph
```

```
INFO: org.hsqldb.persist.Logger: dataFileCache open start
INFO: org.esa.snap.core.datamodel.Product: raster width 3341 not equal to 3340
INFO: org.esa.snap.core.datamodel.Product: raster width 3341 not equal to 3340
```

```
...
```

```
14858 [main] INFO serverStartup - Nc4Iosp: NetCDF-4 C library loaded (jna_path='/home/
↪test/.snap/auxdata/netcdf_natives/8.0.5/amd64', libname='netcdf').
14895 [main] INFO serverStartup - NetcdfLoader: set log level: old=0 new=0
14895 [main] INFO serverStartup - Nc4Iosp: set log level: old=0 new=0
```

```
11%...22%...32%...43%...53%...64%...75%...85%.. done.
```

```
-- org.jblas INFO Deleting /tmp/jblas4294428937644417987/libjblas.so
-- org.jblas INFO Deleting /tmp/jblas4294428937644417987/libgfortran-4.so
-- org.jblas INFO Deleting /tmp/jblas4294428937644417987/libjblas_arch_flavor.so
-- org.jblas INFO Deleting /tmp/jblas4294428937644417987/libquadmath-0.so
-- org.jblas INFO Deleting /tmp/jblas4294428937644417987
INFO:root:0
INFO:root:2021-08-29 20:29:42.545034
INFO:root:combination of default multi temporal speckle filter and user defined graphs␣
↪is not supported yet
```

```
start step 3
start add netcdf information
```

```
INFO:root:Number of scenes found for processing: 2
```

```
start create netcdf stack

Scene 1 of 2
/home/test/Desktop/data4/step2/S1B_IW_GRDH_1SDV_20210706T052551_20210706T052616_027669_
↪034D60_888E_GC_RC_No_Su_Co.nc
```

(continues on next page)

(continued from previous page)

```
Scene 2 of 2
/home/test/Desktop/data4/step2/S1A_IW_GRDH_1SDV_20210707T051822_20210707T051847_038667_
↪04901E_1B6B_GC_RC_No_Su_Co.nc
```

4. View processed data

Load netcdf file with processed data

[11]:
```
import os
print(os.getcwd())
print(output_folder)
```

```
/home/test/Desktop/sar-pre-processing/docs/notebooks
/home/test/Desktop/data4
```

[12]:
```
from netCDF4 import Dataset
import numpy as np

my_example_nc_file = os.path.join(output_folder, 'data4.nc')
data = Dataset(my_example_nc_file, mode='r')
```

View information about dataset

[13]:
```
data
```

[13]:
```
<class 'netCDF4._netCDF4.Dataset'>
root group (NETCDF4 data model, file format HDF5):
    dimensions(sizes): lat(2227), lon(3341), time(2)
    variables(dimensions): float32 time(time), float32 orbitdirection(time), float32␣
↪relorbit(time), float32 satellite(time), float32 lat(lat), float32 lon(lon), float32␣
↪Sigma0_VH(time, lat, lon), float32 Sigma0_VV(time, lat, lon)
    groups:
```

Read data from netcdf file

[14]:
```
data.variables['orbitdirection'][:]
data.variables['time'][:]
lons = data.variables['lon'][:]
lats = data.variables['lat'][:]
data.variables.keys()
```

[14]:
```
dict_keys(['time', 'orbitdirection', 'relorbit', 'satellite', 'lat', 'lon', 'Sigma0_VH',
↪'Sigma0_VV'])
```

[15]:
```
vv = data.variables['Sigma0_VV'][:]
```

Close netcdf file

[16]:
```
data.close()
```

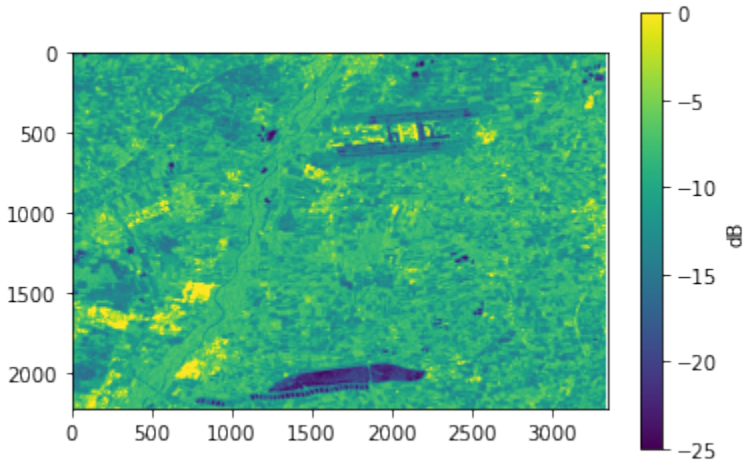
Plot vv polorized data

[17]:
```
%matplotlib inline
from ipywidgets import interactive
import matplotlib.pyplot as plt
import numpy as np

def f(x):
    # Problem: border pixel might be zero or negative
    # pixel eqal or smaller than zero are set to nan
    array = np.copy(vv[x])
    array[array <= 0] = np.nan
    # plot backscatter data in dB scale
    plt.imshow(10*np.log10(array))
    cbar = plt.colorbar()
    cbar.set_label('dB')
    plt.clim(-25, 0)

interactive_plot = interactive(f, x=(0,len(vv)-1))
interactive_plot
```

```
interactive(children=(IntSlider(value=0, description='x', max=1), Output()), _dom_
↪classes=('widget-interact',)...
```



1.3.4 Explanation of config file

```
# Sample config file
#====================
## Necessary parameters
#-----------------------
### Input folder with SAR data (zip format)
input_folder: '/media/test/Desktop/data' # values: string

### Output folder to store the pre-processed data
output_folder: '/media/test/Desktop/data' # values: string

### Location of SNAP's graph-processing-tool
gpt: /home/tweiss/snap/bin/gpt
```

(continues on next page)

(continued from previous page)

```
## Optional parameters
#-----------------------
### Year of interest (only images of the specified year will be processed)
#~~~~~~~~~~~~~~~~~~~
year: 2021 # values: integer

### Area of interest (only images containing the specified year will be processed)
#~~~~~~~~~~~~~~~~~~
region:
  subset: 'yes' # values: 'no' or 'yes'
  ul:
    lat: 48.40 # values: float
    lon: 11.60 # values: float
  lr:
    lat: 48.10 # values: float
    lon: 11.90 # values: float

### Used parameters of multi-temporal speckle filter
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
speckle_filter:
  multi_temporal:
    apply: 'yes' # values: 'no' or 'yes'
    files: '5' # values: integer

### Used parameter of incidence normalization (default angle is 35°)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
normalisation_angle: 35 # values: float

### Single file processing (if filelist contains only one file single_file option is␣
↪automatically set to 'yes')
#~~~~~~~~~~~~~~~~~~~~~~~~
single_file: 'yes' # values: 'no' or 'yes'

### Usage of user defined xml graphs
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
#### Specification if user defined xml graph should be used
use_user_definde_graphs: 'no' # values: 'no' or 'yes'

#### Location of user defined xml files for processing
xml_graph_path: /media/test/Desktop/sar-pre-processing/sar_pre_processing/user_defined_
↪graphs

##### file names of user defined xml graphs
pre_process_step1: example_of_expert_user.xml
```

1.4 Default Pre-Processing Chain

The default pre-processing chain was developed to process Sentinel-1 SLC data to provide geometric and radiometric corrected Sigma nought backscatter values. Among other things the processed data can be used within different radiative transfer models (e.g. Integration Equation Model [1], Oh Model [2], [3], Dubois Model [4], Water Cloud Model [5], Single Scattering Radiative Transfer [6], [7]) to retrieve different land surface and/or vegetation parameters.

1.4.1 Overview

The different preprocessing steps are shown in Fig. 1.1 and Fig. 1.2. Additionally, every processing step is explained in more detail in the following subsections. As it can be seen in Fig. 1.1 and Fig. 1.2 the preprocessing work-flow is split in two main parts. The preprocessing methods in Fig. 1.1 can be applied separately for every image. Whereas the work-flow shown in Fig. 1.2 needs several images which were preprocessed by the different steps presented in Fig. 1.1.

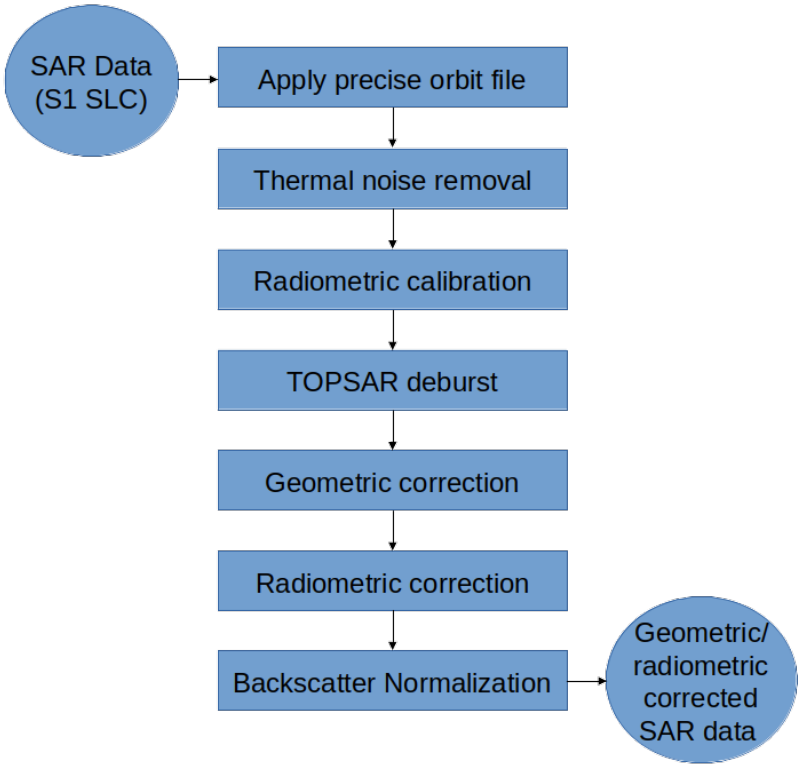


Fig. 1.1: Preprocessing chain showing processing steps to archive geometric and radiometric corrected Sentinel-1 data.

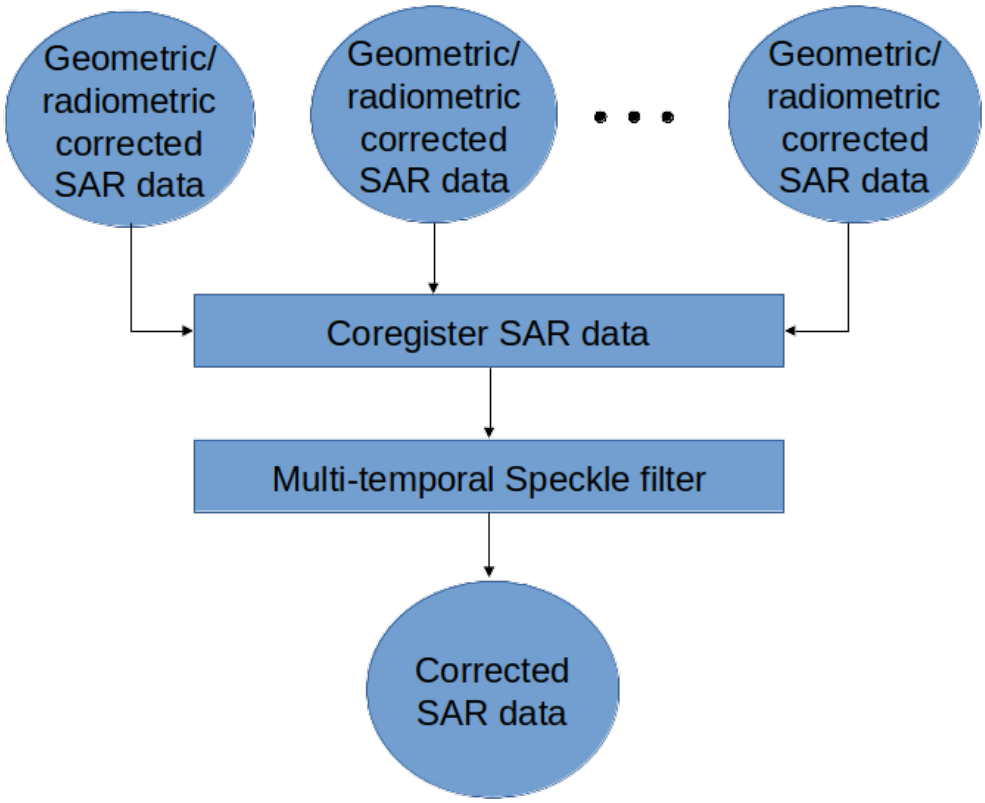


Fig. 1.2: Preprocessing chain showing processing steps to archive co-registered images which are multi-temporal speckle filtered

1.4.2 Sentinel-1 Level-1 SLC data

The preprocessing work-flow of Fig. 1.1 is based on Sentinel-1 Level-1 SLC data. Among some other sources Sentinel-1 data can be downloaded from ESA's Copernicus Open Access Hub (https://scihub.copernicus.eu/).

Sentinel-1 Level-1 SLC data are generated by the operational ESA Instrument Processing Facility (IPF). The SLC products are situated in slant range geometry. The slant range geometry is the natural radar one and is defined by the line-of-sight distance of the radar system to each reflecting object. The SLC product consists of focused SAR data in zero-Doppler orientation. Furthermore, for geo-referencing orbit and attitude information directly provided by the satellite are stored within the SLC product. Moreover the SAR data is corrected for errors caused by the well known azimuth bi-static delay, elevation antenna pattern and range spreading loss [8]. In contrary to Level-1 Ground Range Detected (GRD) products SLC data preserve the real and imaginary part of the backscatter signal and contain therefore also the phase information [8]. The IPF is generating SLC data for all available acquisition modes (StripMap (SM), Interferometric Wide (IW), Extra Wide (EW), and Wave (WV)) of the Sentinel-1 satellites. Further information about Sentinel-1 Level-1 products are gathered in ESA's Sentinel-1 User Handbook [8] available at https://earth.esa.int/documents/247904/685163/Sentinel-1_User_Handbook.

1.4.3 Precise orbit file

Theory / Purpose

During the acquisition of Sentinel-1 data the satellite position is recorded by a Global Navigation Satellite System (GNSS). To assure a fast delivery of Sentinel-1 products orbit information generated by an on-board navigation solution are stored within the Sentinel-1 Level-1 products. The orbit positions are later refined and made available as restituted or precise orbit files by the Copernicus Precise Orbit Determination (POD) Service. The POD products for Sentinel-1 data with given accuracy and availability after data acquisition are listed in Table 1.1.

Table 1.1: Accuracy specification for Sentinel-1 POD products [9]

Mission	POD Product	Accuracy	Latency
Sentinel-1	Restituted Orbit File	< 10 cm	3 hours
	Precise Orbit Ephemerides (POE) Orbit file	< 5 cm	20 days
	Attitude Restituted Data	< 0.005 deg	20 days

Precise orbit information can have a high influence on the quality of several preprocessing steps especially e.g. for the geo-referencing of the data. Therefore, it is always preferable to use the most accurate orbit information that is available.

Practical implementation

Since the preprocessing for the MULTIPLY project doesn't depend on near-real-time data the precise orbit file (available within 20 days) is used to update the orbit and velocity information within the Sentinel-1 SLC product. Therefore the operator "Apply Orbit Correction" of SNAP S1TBX toolbox is used.

Input:

- Sentinel-1 SLC IW image (downloaded from Copernicus Open Access Hub)
- Precise orbit file (automatic download by SNAP S1TBX)

Output:

- Sentinel-1 SLC IW image with updated orbit information

1.4.4 Thermal noise removal

Theory / Purpose

Thermal noise is caused by the background energy of a SAR receiver and independent from the received signal power. Like some other noise factors thermal noise appears randomly over the entire image. But in contrary to quantization noise like speckle, which is connected to the signal power, thermal noise is hardly noticeable. Therefore, high impact of thermal noise on the quality of the data is especially given in areas like calm lakes, rivers and other with a low mean signal response detected by the SAR system. For the purpose of correction the IPF is calculating a thermal noise Look up Table (LUT) which is stored within the Sentinel-1 Level-1 product. More information about the calculation of the thermal noise for Sentinel-1 is given in [10].

Practical implementation

The "Thermal Noise Removal" operator of SNAP S1TBX software is used to remove the thermal noise which is stored within a LUT within Sentinel-1 Level-1 products. Thermal noise removal can only applied on backscatter intensity therefore the phase information of the SLC data get lost.

Input:

- Sentinel-1 SLC IW image with updated orbit information

Output:

- Sentinel-1 SLC Intensity corrected by thermal noise

1.4.5 Radiometric calibration

Theory / Purpose

Sentinel-1 Level-1 products are not radiometric corrected by default. However, for the quantitative use of SAR images a radiometric calibration of radar reflectivity (stored as Digital Numbers (DN) within Sentinel-1 Level-1 products) to physical units (radar backscatter) is essential. Otherwise a comparison of SAR images from different sensors or even the same sensor for different acquisition dates or different acquisition modes is not possible. To apply a radiometric calibration a Calibration Annotation Data Set (CADS) with four Look Up Tables (LUTs) are provided within the Sentinel-1 Level-1 products by Sentinel-1 Instrument Processing Facility (IPF). The four LUTs are used to convert DN to sigma naught, beta naught and gamma or vice versa. More information about the radiometric calibration is given in [11].

Practical implementation

The "Radiometric Calibration" operator of SNAP S1TBX software is used to perform the conversion of DN to radar backscatter. In our case the output radar backscatter information is calibrated in Sigma naught.

Input:

- Sentinel-1 SLC Intensity corrected by thermal noise

Output:

- Sigma naught calibrated radar backscatter

1.4.6 TOPSAR Deburst

Theory / Purpose

Sentinel-1 Level-1 SLC images acquired in IW or EW swath mode consists of one image per swath and polarisation. IW products are made up of three swaths which means three images for single polarisation and six images for dual polarisation. EW products are made up of five swaths which means five images for single polarisation and ten images for dual polarisation. The sub-swath images consists of different bursts which are all processed as separate images. The different bursts are stored in one single image whereby each burst is separated by a black-filled demarcation [8]. For the usage of Sentinel-1 Level-1 SLC data only one sub-swath can be extracted or several/all sub-swath can be combined to one image with fluent transitions between the sub-swaths. More detailed information are provided in [8], [12] and [13].

Practical implementation

The "TOPSAR-Deburst" operator of SNAP S1TBX software is used to merge all sub-swath to retrieve one fluent image.

Input:

- Sigma naught calibrated radar backscatter (with different sub-swath)

Output:

- Sigma naught calibrated radar backscatter (with fluent transitions)

1.4.7 Geometric correction

Theory / Purpose

An important part of the preprocessing chain is the geometric terrain correction. The geometric correction is a conversion of the Sentinel-1 SLC data from slant range geometry into a map coordinate system. Due to the acquisition geometry of the SAR different topographical distortions like foreshortening, layover or shadowing effects occur. The appropriate way to correct these distortions is the Range-Doppler approach. The method needs information about the topography (normally provided by a Digital Elevation Model (DEM)) as well as orbit and velocity information from the satellite (stored within Sentinel-1 SLC product) to correct the mentioned distortions and derive a precise geolocation for each pixel of the image.

Practical implementation

A geometric correction of the input data is performed by using the "Range Doppler Terrain Correction" method implement in SNAP's S1TBX software. Data from the Shuttle Radar Topography Mission (SRTM) with a resolution of 1-arc second (30 meters) is used for the necessary DEM.

Input:

- Sigma naught calibrated radar backscatter (with fluent transitions)
- SRTM data with 1-arc second resolution (automatic download by SNAP S1TBX)

Output:

- Geometric corrected sigma naught calibrated radar backscatter (Map Projection WGS84)
- Incidence angle from ellipsoid
- Local incidence angle (based on SRTM)

1.4.8 Radiometric correction

Theory / Purpose

For the conversion of Sentinel-1 backscatter values to sigma or gamma naught, LUT's stored within the Sentinel-1 product are used (see *Radiometric calibration*). For the creation of the LUT's Sentinel-1 IPF is using an incidence angle of an ellipsoid inflated earth model [11]. Therefore, the local terrain variation within the image and their radiometric impact on the backscatter is considered insufficiently. A simple and widely used practice to consider the radiometric impact due to local terrain variations represents the approach to use the local incidence angle instead of the ellipsoid one [14]. The radiometric corrected backscatter σ^0_{NORLIM} used by kellndorfer_toward_1998 et al. [14] can be calculated as

$$\sigma^0_{NORLIM} = \sigma_{Ell} \frac{sin\theta_{LIA}}{sin\theta_{Ell}} \tag{1.1}$$

with θ_{LIA} as the local incidence angle, θ_{Ell} as the ellipsoid incidence angle used by IPF and the radar backscatter σ_{Ell} calculated by using LUT's provided by IPF.

Practical implementation

Within the "Range Doppler Terrain Correction" method of SNAP's S1TBX software the radiometric normalisation approach of kellndorfer_toward_1998 et al. [14] is implemented as a additional option. Unfortunately, the SNAP internal option can not be used with our kind of data. Therefore, normalisation after kellndorfer_toward_1998 et al [14] is done by coding the equations within the "BandMath" operator of SNAP's S1TBX. The used local incidence angle is provided by the previous applied "Range Doppler Terrain Correction" and therefore the local incidence angle is based on the SRTM data.

Input:

- Geometric corrected sigma naught calibrated radar backscatter (Map Projection WGS84)
- Incidence angle from ellipsoid
- Local incidence angle (based on SRTM)

Output:

- Radiometric and geometric corrected sigma naught calibrated radar backscatter (Map Projection WGS84)

1.4.9 Backscatter normalisation

Theory / Purpose

Beside the previously discussed geometric and radiometric distortions some other specific backscattering coefficient variations within the range direction of the image are caused by the image geometry of the SAR sensor. The backscattered energy of an illuminated area has not only a dependency on the area itself but also on the incidence angle. This means, backscatter values of a specific area with a small incidence angle return higher backscatter values then data of the same area acquired with a higher incidence angle. Incidence angle induced variations not only occur inside one image but also between images form different sensors as well as within one sensor through different acquisition geometries or different tracks or orbits. For a usage of Sentinel-1A and 1B time-series acquired with different orbits and/or different tracks and therefore most likly a high change between the incidence angles a backscatter normalisation is vital. A often and widely used technique to minimize backscatter variations caused by the incidence angle is the cosine correction [15]. The cosine correction is based on the Lambert's law for optics. Therefore, under the assumption that the backscattered energy in the upper hemisphere follows a cosine law and also the radiation variability has a cosine dependency, the received backscatter $\sigma^0_{\theta_i}$ and its dependency on the incidence angle can be written as

$$\sigma^0_{\theta_i} = \sigma^0_0 cos^n(\theta_i) \tag{1.2}$$

with a weighting factor n and the incidence angle independent backscatter σ_0^0. With the cosine correction the backscatter of the Sentinel-1 products can therefore normalised to a reference angle θ_{ref} with

$$\sigma_{ref}^0 = \frac{\sigma_{\theta_i}^0 cos^n(\theta_{ref})}{cos_{\theta_i}^n} \tag{1.3}$$

Studies show that the weighting factor n is dependent on the roughness [16] and therefore the backscatter variations can vary with different land cover types. A schematic illustration of the backscatter variations considering the incidence angle is given in Fig. 1.3.

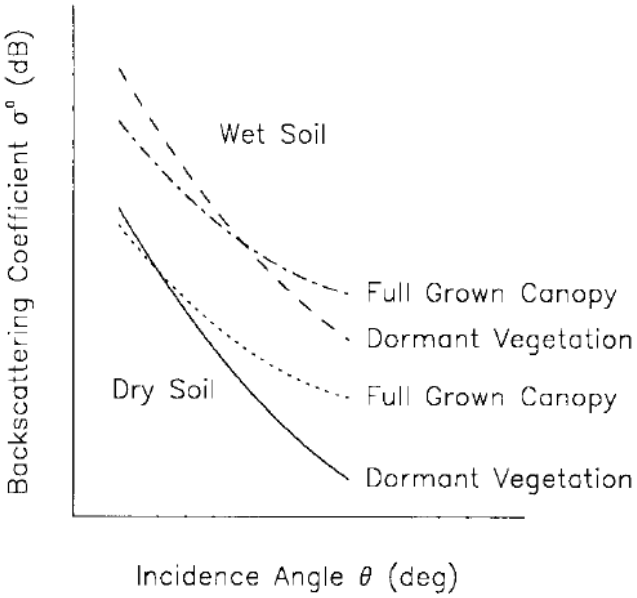


Fig. 1.3: Illustration of the backscatter variations considering the incidence angle dependency [17].

Practical implementation

The backscatter normlisation is applied by coding (1.3) in SNAP's S1TBX operator "BandMaths". As default a reference angle of 37,55° (average incidence angle for IW swath mode [10]) and a weighting factor of 2 (standard value) is specified. Through a configuration file the user can replace the default values for the reference angle and weighting factor to probably more suitable values of their specific applications.

Input:

- Radiometric and geometric corrected sigma naught calibrated radar backscatter (Map Projection WGS84)
- reference angle (default is 35°)
- weighting factor (default is 2)

Output:

- Radiometric and geometric corrected sigma naught calibrated radar backscatter values normalised to reference angle (Map Projection WGS84)

1.4.10 Co-registration

Theory / Purpose

For time-series analysis especially when applying a *Multi-temporal speckle filter* the SAR image has to be co-registered. The co-registration is a method to get every image of the time-series on the same grid and also the pixel resolution.

Practical implementation

The co-registration as a requirement for the *Multi-temporal speckle filter* is accomplished by the "Co-Registration" operator within SNAP's S1TBX. The "Co-Registration" operator in SNAP is defined as a completely automatic process. The operator consists of a stack creation (collocating master and slave image), a cross correlation (allignment between master ans slave image) and a warp (resamples pixels from the slave image to pixels of the master image).

Input:

- Master image
- Slave image(s)

Output:

- Co-registered images

1.4.11 Multi-temporal speckle filter

Theory / Purpose

A characteristic of images acquired by a SAR system is the visibility of random noise which look like "salt and pepper" within the image and is called speckle. The appearance of speckle is caused by the interferences of coherent echoes from individual scatterers within one pixel [18].The presence of speckle degrades the quality of the image and therefore it makes the interpretation of the SAR data more difficult. Over the years several approaches for speckle reduction were developed. They are mainly based on either multi-looking or filtering methods. Different filtering approaches like Frost, Lee etc. can be applied as a single or multi-temporal speckle filter. First findings with Sentinel-1 data show that a multi-temporal speckle filter provides better results in form of speckle reduction and resolution preservation then a single speckle filter. A major advantage for the usage of a multi-temporal speckle filter on Sentinel-1 data is the high temporal resolution availability. Nevertheless, more detailed studies on analysing the impact of different multi-temporal speckle filters on the retrieval of bio- and geophysical parameters from Sentinel-1 data are still lacking. Anyway, a usage of a multi-temporal filter significantly reduces the speckle and is therefore a essentinal part of our preprocessing chain.

Practical implementation

For the speckle reduction the "Multi-temporal Speckle Filter" operator within SNAP's S1TBX software is used. As default, 7 temporally consecutive images are used within the "Multi-temporal Speckle Filter" whereby the target image is temporally situated in the middle. The applied filter is a Lee filter with a spatial window size of 5x5 pixels, a sigma of 0.9, and a target window size of 3x3 pixels. The spatial averaging over pixel has a significant influence on spatial resolution information loss of the image. Therefore, the averaging pixel size might change during the project. If the image consists of two polarisations the filter is applied on each polarisation separately. The practical implementation in case of filter type, used polarisation, number of used images etc. may change with more experience of applying multi-temporal speckle filters and the occurring results.

Input:

- x co-registered images (can be specified within configuration file)

Output:

- speckle filtered images

1.4.12 Folder and data creation during pre-processing steps

Creation of

- **step1 (folder)**
  - temporary results after applying processing steps shown in Fig. 1.1
- **step2 (folder)**
  - co-registered images of step1
- **step3 (folder)**
  - final results
- foldername.nc (final netcdf stack file)

within specified output folder (config file)

1.4.13 Output layers of final netcdf stack file

Output layer of default pre-processing chain

- theta (local incidence angle)
- sigma0_vv_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vh_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vv_multi (multi speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vh_multi (multi speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vv_norm_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vh_norm_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vv_norm_single (multi speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vh_norm_single (multi speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)

1.4.14 Abbreviations and values within netcdf stack file

Abbreviation within variable names

- theta = local incidence angle
- sigma0 = radiometric and geometric corrected sigma nought backscatter
- vv = VV polarization
- vh = VH polarization

- single = single speckle filter was applied
- multi = multitemporal speckle filtered
- norm = backscatter was normalized to a specific incidence angle

Values of specific variables

- **orbitdirection**
  - 0 = Ascending
  - 1 = Descending
- **relorbit**
  - number of relative orbit
- **satellite**
  - 0 = Sentinel-1 A
  - 1 = Sentinel-1 B
- **name tags of processed data**
  - theta (local incidence angle)
  - sigma0 (radiometric and geometric corrected sigma nought backscatter)
  - vv (polarization vv)
  - vh (polarization vh)
  - single (single speckle filtered)
  - multi (multi temporal speckle filtered)
  - norm (backscatter normalized to a specific incidence angle)

References

1.5 Technical documentation

1.5.1 Sar-Pre-Processing

Wrapper module to launch preprocessor

class sar_pre_processing.sar_pre_processor.**PreProcessor**(***kwargs*)
Bases: object

static pre_process()

class sar_pre_processing.sar_pre_processor.**SARPreProcessor**(***kwargs*)
Bases: *sar_pre_processing.sar_pre_processor.PreProcessor*

add_netcdf_information()
Add information from original S1 image to processed NetCDF file. - update date information (wrong date information were stored due to coregistration process) - orbitdirection (ASCENDING, DESCENDING) - relative orbit - Satellite (S1A, S1B) - Frequency

create_netcdf_stack(*filename: Optional[str] = None*)
create one NetCDF stack file of pre-processed data Orbitdirection: '0 = Ascending, 1 = Descending' Satellite: '0 = Sentinel 1A, 1 = Sentinel 1B'

create_processing_file_list()
create a list with all to be processed file names

pre_process_step1()
Pre-process Sentinel-1 data - Default processing chain of SenSARP:

Pre-process S1 SLC data with SNAP's GPT

1) apply precise orbit file
2) thermal noise removal
3) calibration
4) TOPSAR-Deburst
5) Geometric Terrain Correction
6) Radiometric Correction (after kellndorfer et al.)
7) backscatter normalisation on specified angle in config file (based on Lambert's Law)

Output layers:

- longitude
- latitude
- localIncidenceAngle
- projectedLocalIncidenceAngle
- incidenceAngleFromEllipsoid
- elevation
- Sigma0_VV (VV-polarized Sigma nought backscatter no radiometric correction applied)
- Sigma0_VH (VH-polarized Sigma nought backscatter no radiometric correction applied)
- sigma0_vv_kelln (VV-polarized Sigma nought backscatter including radiometric correction after Kellndorfer)
- sigma0_vh (VH-polarized Sigma nought backscatter including radiometric correction after Kellndorfer)
- sigma0_vv_kelln_normalisation (VV-polarized Sigma nought backscatter including radiometric correction after Kellndorfer and normalization to one specific incidence angle)
- sigma0_vh_kelln_normalisation (VH-polarized Sigma nought backscatter including radiometric correction after Kellndorfer and normalization to one specific incidence angle)

- use processing chain of expert user

pre_process_step2()
pre_process_step1 has to be done first

pre_process_step2 only useful for processing time series data

1) co-register pre-processed data

!!! all files will get metadata of the master image !!! That is how SNAP does it! Metadata will be corrected within netcdf output files at the end of the preprocessing chain (def add_netcdf_information)

pre_process_step3()
pre_process_step1 and/or 2 has to be done first

processing time series data:

- apply multi-temporal speckle filter and single speckle filter

processing single image:

- apply single speckle filter

Output layers:

- theta (local incidence angle)
- sigma0_vv_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vh_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vv_multi (multi speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vh_multi (multi speckle filtered radiometric and geometric corrected sigma nought backscatter)
- sigma0_vv_norm_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vh_norm_single (single speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vv_norm_single (multi speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)
- sigma0_vh_norm_single (multi speckle filtered radiometric and geometric corrected sigma nought backscatter normalized to a specific incidence angle)

1.5.2 File list for Sar-Pre-Processing

Create List of SAR data which will be processed by sar_pre_processer module

class sar_pre_processing.file_list_sar_pre_processing.**SARList**(***kwargs*)
Bases: object

Object for creation of file list for preprocessing of Sentinel-1 data based on configuration file

create_list(***kwargs*)
Create file list for further processing

Filter option via config file - year - area of interest

Checking for

- double processed data by ESA
- area of interest contained in two tiles of same swath

1.5.3 netcdf-stack

class sar_pre_processing.netcdf_stack.**NetcdfStackCreator**(**kwargs)
Bases: object

Create NetCDF stack

create_netcdf_stack()

stacking()
stack all files into a new netcdf file created by function _create_empty_netcef_file

1.5.4 Attribute Dict

class sar_pre_processing.attribute_dict.**AttributeDict**(**entries)
Bases: object

A class to convert a nested Dictionary into an object with key-values accessibly using attribute notation (AttributeDict.attribute) instead of key notation (Dict["key"]). This class recursively sets Dicts to objects, allowing you to recurse down nested dicts (like: AttributeDict.attr.attr)

add_entries(**entries)

add_entry(*key*, *value*)

has_entry(*key*)

1.6 Authors

- Thomas weiß <"weiss.thomas@lmu.de">
- Tonio Fincke <"tonio.fincke@brockmann-consult.de">

1.7 Change Log SenSARP

1.7.1 [0.1] - 2021-04-20

initial version

1.7.2 [0.2] - 2022-01-18 (JOSS Paper)

JOSS Paper

- extensive revision

Documentation

- add installation guide using virtualenv and pip
- revise introduction
- add Statement of need
- add explanations of created files and used abbreviations
- name change from "MULTIPLY SAR pre-processing" to "SenSARP"
- **add notebooks (use cases)**
  - default_process_single_image.ipynb
  - default_process_time_series.ipynb
  - use_user_defined_graphs.ipynb
- remove notebook running_test_application.ipynb

Software

- add more explanation to config file
- add more documentation to functions of class SARPreProcessor
- add functionality that to user defined xml-graphs can be used
- add functionality that a single file can be processed
- class NetcdfStackCreator was partly rewritten
- functionality of shell file (solve_projection_problem.sh) was included within python code
- bug fixes

1.8 Licence

1.8.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program–to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running

the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in

which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year>  <name of author>

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see https://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read https://www.gnu.org/licenses/why-not-lgpl.html.

CHAPTER

TWO

CREDITS

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 687320.

BIBLIOGRAPHY

[1] A.K. Fung, Z. Li, and K.S. Chen. Backscattering from a randomly rough dielectric surface. *IEEE Transactions on Geoscience and Remote Sensing*, 30(2):356–369, March 1992. URL: http://ieeexplore.ieee.org/document/134085/ (visited on 2019-10-22), doi:10.1109/36.134085.

[2] Y. Oh, K. Sarabandi, and F.T. Ulaby. An empirical model and an inversion technique for radar scattering from bare soil surfaces. *IEEE Transactions on Geoscience and Remote Sensing*, 30(2):370–381, March 1992. doi:10.1109/36.134086.

[3] Yisok Oh. Quantitative retrieval of soil moisture content and surface roughness from multipolarized radar observations of bare soil surfaces. *IEEE Transactions on Geoscience and Remote Sensing*, 42(3):596–601, March 2004. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. doi:10.1109/TGRS.2003.821065.

[4] P.C. Dubois, J. van Zyl, and T. Engman. Measuring soil moisture with imaging radars. *IEEE Transactions on Geoscience and Remote Sensing*, 33(4):915–926, July 1995. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. doi:10.1109/36.406677.

[5] E. P. W. Attema and Fawwaz T. Ulaby. Vegetation modeled as a water cloud. *Radio Science*, 13(2):357–364, March 1978. Conference Name: Radio Science. doi:10.1029/RS013i002p00357.

[6] **missing publisher in ulaby_microwave_2014**

[7] Thomas Weiß, Thomas Ramsauer, Alexander Löw, and Philip Marzahn. Evaluation of Different Radiative Transfer Models for Microwave Backscatter Estimation of Wheat Fields. *Remote Sensing*, 12(18):3037, January 2020. Number: 18 Publisher: Multidisciplinary Digital Publishing Institute. URL: https://www.mdpi.com/2072-4292/12/18/3037 (visited on 2020-09-29), doi:10.3390/rs12183037.

[8] Sentinel-1 Team. Sentinel-1 User Handbook. technical GMES-S1OP-EOPG-TN-13-0001, ESA, 2013.

[9] Sentinels POD team. Sentinels POD Service File Format Specifications. technical GMES-GSEG-EOPG-FS-10-0075, ESA, 2016.

[10] Matthieu Bourbigot, Harald Johnson, and Riccardo Piantanida. Sentinel-1 Product Definition. technical S1-RS-MDA-53-7440, ESA, 2015.

[11] Nuno Miranda and Peter Meadows. Radiometric Calibration of S-1 Level-1 Products Generated by the S-1 IPF. technical ESA-EOPG-CSCOP-TN-0002, ESA, 2015.

[12] D. D'Aria, F. De Zan, D. Giudici, A. Monti Guarnieri, and F. Rocca. Burst-mode SARs for wide-swath surveys. *Canadian Journal of Remote Sensing*, 33(1):27–38, January 2007. Publisher: Taylor & Francis _eprint: https://doi.org/10.5589/m07-008. URL: https://doi.org/10.5589/m07-008 (visited on 2021-08-20), doi:10.5589/m07-008.

[13] F. De Zan and A. Monti Guarnieri. TOPSAR: Terrain Observation by Progressive Scans. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2352–2360, September 2006. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. doi:10.1109/TGRS.2006.873853.

[14] J.M. Kellndorfer, L.E. Pierce, M.C. Dobson, and F.T. Ulaby. Toward consistent regional-to-global-scale vegetation characterization using orbital SAR systems. *IEEE Transactions on Geoscience and Remote Sensing*, 36(5):1396–1411, September 1998. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. doi:10.1109/36.718844.

[15] Fawwaz T. Ulaby. *Microwave remote sensing: active and passive. Vol.2, Radar remote sensing and surface scattering and emission theory*. Remote sensing. Artech House, Norwood, MA, 1986. ISBN 978-0-89006-191-6.

[16] Juan Pablo Ardila, Valentyn Tolpekin, and Wietske Bijker. Angular Backscatter Variation in L-Band ALOS ScanSAR Images of Tropical Forest Areas. *IEEE Geoscience and Remote Sensing Letters*, 7(4):821–825, October 2010. Conference Name: IEEE Geoscience and Remote Sensing Letters. doi:10.1109/LGRS.2010.2048411.

[17] W. Wagner, G. Lemoine, M. Borgeaud, and H. Rott. A study of vegetation cover effects on ERS scatterometer data. *IEEE Transactions on Geoscience and Remote Sensing*, 37(2):938–948, March 1999. Conference Name: IEEE Transactions on Geoscience and Remote Sensing. doi:10.1109/36.752212.

[18] Iain H. Woodhouse. *Introduction to Microwave Remote Sensing*. CRC Press, Boca Raton, 2006. ISBN 978-1-315-27257-3. doi:10.1201/9781315272573.

PYTHON MODULE INDEX

S

sar_pre_processing.attribute_dict, 54
sar_pre_processing.file_list_sar_pre_processing, 53
sar_pre_processing.netcdf_stack, 54
sar_pre_processing.sar_pre_processor, 51

INDEX

A

add_entries() (*sar_pre_processing.attribute_dict.AttributeDict method*), 54
add_entry() (*sar_pre_processing.attribute_dict.AttributeDict method*), 54
add_netcdf_information() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 51
AttributeDict (*class in sar_pre_processing.attribute_dict*), 54

C

create_list() (*sar_pre_processing.file_list_sar_pre_processing.SARList method*), 53
create_netcdf_stack() (*sar_pre_processing.netcdf_stack.NetcdfStackCreator method*), 54
create_netcdf_stack() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 52
create_processing_file_list() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 52

H

has_entry() (*sar_pre_processing.attribute_dict.AttributeDict method*), 54

M

module
    sar_pre_processing.attribute_dict, 54
    sar_pre_processing.file_list_sar_pre_processing, 53
    sar_pre_processing.netcdf_stack, 54
    sar_pre_processing.sar_pre_processor, 51

N

NetcdfStackCreator (*class in sar_pre_processing.netcdf_stack*), 54

P

pre_process() (*sar_pre_processing.sar_pre_processor.PreProcessor static method*), 51
pre_process_step1() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 52
pre_process_step2() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 52
pre_process_step3() (*sar_pre_processing.sar_pre_processor.SARPreProcessor method*), 53
PreProcessor (*class in sar_pre_processing.sar_pre_processor*), 51

S

sar_pre_processing.attribute_dict module, 54
sar_pre_processing.file_list_sar_pre_processing module, 53
sar_pre_processing.netcdf_stack module, 54
sar_pre_processing.sar_pre_processor module, 51
SARList (*class in sar_pre_processing.file_list_sar_pre_processing*), 53
SARPreProcessor (*class in sar_pre_processing.sar_pre_processor*), 51
stacking() (*sar_pre_processing.netcdf_stack.NetcdfStackCreator method*), 54