SenSARP: A Python package for pre-processing Sentinel-1 SLC data with the Sentinel-1 Toolbox

Thomas Weiß[1] **and Tonio Fincke**[2]

1 Department of Geography, Ludwig-Maximilians-Universität München, 80333 Munich, Germany **2** Brockmann Consult GmbH, 21029 Hamburg, Germany;

Summary

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 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.

This python package generates a file list of to be processed Sentinel-1 images (already downloaded and stored in a specific folder) based on different user defined criteria (specific year, area of interest). Additionally, specific cases of repeatedly processed data are handled, as sometimes Sentinel-1 data were initially processed multiple times and stored under similar names on the Copernicus Open Access Hub. Also, cases where Sentinel-1 data within the user-defined area of interest might be stored in consecutive images are considered.

Based on the generated file list the python package applies a pre-processing chain to Sentinel-1 Single Look Complex (SLC) time series data to generate radiometrically and geometrically corrected Sigma nought backscatter values. Furthermore, the time series images are co-registered and additional output files of multi-temporal speckle filtered data are generated. In addition, a single speckle filter instead of a multi-temporal one can be applied as well and the output will be stored as a separate layer. To pre-process the images, the python package uses the GPF (Graph Processing Framework) of the SeNtinel Application Platform (SNAP) and the operators provided by the Sentinel-1 Toolbox (in version 8.0.3). The Sentinel Toolbox is available for download at step.esa.int, its source code is available in the senbox-org organization on Github. Each of these operators performs a pre-processing step. The operators can be chained together to form a graph, which is used by the python package to run on the Sentinel-1 data using the GPF. The graphs are stored in xml-files. Users may change the graphs by modifying the files directly or via the Sentinel Toolbox. User Guides to show how the GPF can be used are provided here: https://senbox.atlassian.net/wiki/spaces/SNAP/pages/70503053/Processing.

After the pre-processing the resulting radiometrically and geometrically corrected images are stored for further usage within a NetCDF4 stack file. Among other applications the processed

44 images can be used e.g. for flood risk analysis, monitoring land cover changes, monitoring
45 global food security or estimation of land surface parameters. In the future many more new
46 products and operational third party services based on consistent Sentinel-1 time series might
47 be developed.

48 This python package was developed within the Horizon 2020 project called MULTIscale SEN-
49 TINEL land surface information retrieval Platform (MULTIPLY) (http://www.multiply-h2020.
50 eu/, https://cordis.europa.eu/project/id/687320, https://multiply.obs-website.eu-de.otc.
51 t-systems.com). Furthermore, data processed by this package is used within Sentinel-Synergy-
52 Study S3 project (https://www.researchgate.net/project/Sentinel-Synergy-Study-S3). In
53 addition, the python code was used to process Sentinel-1 time series images for the detection
54 and analysis of temporary flooded vegetation (Tsyganskaya et al., 2018, 2019) and for the
55 evaluation of different radiative transfer models for microwave backscatter estimation of
56 wheat fields (Weiß et al., 2020).

Acknowledgements

References

63 Tsyganskaya, V., Martinis, S., & Marzahn, P. (2019). Flood Monitoring in Vegetated Areas
64   Using Multitemporal Sentinel-1 Data: Impact of Time Series Features. *Water*, *11*(9),
65   1938. https://doi.org/10.3390/w11091938

66 Tsyganskaya, V., Martinis, S., Marzahn, P., & Ludwig, R. (2018). Detection of Temporary
67   Flooded Vegetation Using Sentinel-1 Time Series Data. *Remote Sensing*, *10*(8), 1286.
68   https://doi.org/10.3390/rs10081286

69 Weiß, T., Ramsauer, T., Löw, A., & Marzahn, P. (2020). Evaluation of Different Radiative
70   Transfer Models for Microwave Backscatter Estimation of Wheat Fields. *Remote Sensing*,
71   *12*(18), 3037. https://doi.org/10.3390/rs12183037