

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

Thomas Weiß<sup>1</sup> and Tonio Fincke<sup>2</sup>

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

DOI: [10.21105/joss.0XXXX](https://doi.org/10.21105/joss.0XXXX)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Editor Name](#) ↗

Submitted: 01 January XXXX

Published: 01 January XXXX

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## 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 by ESA and the European Commission. 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). 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.

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

This python package was developed within the Horizon 2020 project called MULTIscale SENTINEL land surface information retrieval Platform (MULTIPLY) (<http://www.multiply-h2020.eu/>, <https://cordis.europa.eu/project/id/687320>). Furthermore, data processed by this package is used within Sentinel-Synergy-Study S3 project (<https://www.researchgate.net/project/Sentinel-Synergy-Study-S3>). In addition, the python code was used to process Sentinel-1 time series images for the detection and analysis of temporary flooded vegetation (Tsyganskaya et al., 2018, 2019) and for the evaluation of different radiative transfer models for microwave backscatter estimation of wheat fields (Weiß et al., 2020).

## Acknowledgements

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. We would like to thank Alexander Löw and Philip Marzahn for guiding discussions that lead to this publication. We also would like to thank Thomas Ramsauer for discussions and suggestions.

## References

- Tsyganskaya, V., Martinis, S., & Marzahn, P. (2019). Flood Monitoring in Vegetated Areas Using Multitemporal Sentinel-1 Data: Impact of Time Series Features. *Water*, 11(9), 1938. <https://doi.org/10.3390/w11091938>
- Tsyganskaya, V., Martinis, S., Marzahn, P., & Ludwig, R. (2018). Detection of Temporary Flooded Vegetation Using Sentinel-1 Time Series Data. *Remote Sensing*, 10(8), 1286. <https://doi.org/10.3390/rs10081286>
- Weiß, T., Ramsauer, T., Löw, A., & Marzahn, P. (2020). Evaluation of Different Radiative Transfer Models for Microwave Backscatter Estimation of Wheat Fields. *Remote Sensing*, 12(18), 3037. <https://doi.org/10.3390/rs12183037>