Journal of
open research software

Saavedra, P and Simmer, C 2018 An Octave/MATLAB® Interface for Rapid Processing of SMOS L1C Full Polarization Brightness Temperature. *Journal of Open Research Software,* 6: 2. DOI: https://doi.org/10.5334/jors.165

SOFTWARE METAPAPER

An Octave/MATLAB® Interface for Rapid Processing of SMOS L1C Full Polarization Brightness Temperature

Pablo Saavedra and Clemens Simmer

Meteorological Institute, University of Bonn, DE

Corresponding author: Pablo Saavedra, Research Scientist (pablosaa@uni-bonn.de)

A tool to process the SMOS microwave radiometer level 1C polarized brightness temperatures data product has been developed. The SMOS L1C science product contains the dual and full (Stokes vector) polarization brightness temperatures at L-band for multiple incidence angles. In order to use the L1C product, the measurements are processed by a number of procedures including radio frequency interference (RFI) filters, conversion of the polarization plane from the antenna (X- & Y-pol) to the Earth's surface frame (H- & V-pol), and averaging to fixed classes of incidence angles. The software allows for the processing of data for the entire daily half-orbit product, or for specific regions of interest, and can be adapted as a bash-job to process a large number of data files e.g. for time series analysis. This paper describes the tool which was developed in GNU C++ with the capability to be compiled as MEX function to work with Octave or MATLAB® without any source code adjustment.

Keywords: SMOS; Brightness Temperature; Polarization; L-band; Octave; Matlab
Funding statement: *Deutsche Forschungsgemeinschaft* DFG under grant number SI 606/24-1.

(1) Overview

Introduction

The Soil Moisture Ocean Salinity (SMOS) mission is a European Space Agency (ESA) earth observation satellite carrying the Microwave Imaging Radiometer using Aperture Synthetic (MIRAS) instrument. MIRAS is a passive 2-D interferometric radiometer operating at a frequency of 1.413 GHz (21 cm, L-band) and is used to measure the microwave emission from Earth's surface as brightness temperature (*TB*), from which global maps of upper-level soil moisture and ocean salinity can be estimated [10].

The SMOS satellite follows a sun synchronous orbit (dawn/dusk at 6:00/18:00), and measurements are performed for a range of incidence angles between 0° and 60° across a swath of approximately 1000 km width with a spatial resolution ranging from 35 to 50 km depending on incidence angle. The MIRAS interferometer measures brightness temperatures in dual and full polarization (4 Stokes vector) mode.

The instrument data product comprises three level-1 sub-products:

[L1A] calibrated visibilities between individual antenna receivers (i.e. the interferometric measurements from the sensor including the redundant visibilities), combined per integration time of 1.2 s (snapshot) in full polarization and prior the application of image reconstruction, provided in pole-to-pole (half orbit) time-based segments,

[L1B] brightness temperature product as output of the image reconstruction of the SMOS observations, which consist of the Fourier components of brightness temperatures in the antenna polarization reference frame (XY polarization, hereafter),

[L1C] multi-incidence angle brightness temperatures at the top of the atmosphere (XY polarization), geo-located in an equal-area grid. The L1C product comprises separate datasets available for ocean and land pixels with information available per pixel and per snapshot. An addition a so-called browse product contains the brightness temperatures averaged for a fixed incidence angle of 42.5°.

This document describes the analysis and processing of the L1C scientific data product, which consists of half-orbit daily archives with two separated physical files for each half-orbit: the product header file (HDR) and the data block binary file (DBL).

The Earth surface is viewed by SMOS in a snapshot-like configuration, which is restructured in the L1C product on an equal area Digital Global Grid (DGG) [13] with pixels based on the Icosahedral Snyder Equal Area (ISEA) projection at a spatial separation of about 15 km [11].

The SMOS L1C data product arranges the brightness temperatures sequentially for the real components of the XX, XY or YY polarizations, along with the same polarization sequence for the imaginary polarization components. The imaginary part is always zero for the single polarization components (e.g. XX or YY polarization). **Figure 1** illustrates typical SMOS L1C half-orbit measurements stored in a DBL data file.

In order to use SMOS measurements in further studies for soil moisture or ocean salinity retrievals and for data assimilation, the L1C brightness temperature needs to be transformed from the antenna reference frame (XY) into the Earth's surface reference frame (HV polarization, hereafter). This in done by considering the amount of rotation suffered by the polarization plane due to geometric and Faraday rotation.

Although the SMOS radiometer operates at L-band, which is a protected frequency band, observations are partly affected by Radio Frequency Interference (RFI) from various sources like radars and TV transmitters. Therefore RFI detection and mitigation for L1C brightness temperatures are important procedures, which need to be applied before brightness temperatures are used for retrievals of soil moisture and ocean salinity. The magnitude at which RFI affect the data, depends on the orbit (ascending or descending), polarization, and observation incidence angle. E.g., at higher incidence angles the pixel footprint is larger, which increases the impact of RFI compared to observations at nadir.

Some tools for polarization reference frame transformation and RFI mitigation have already been developed (e.g. SMOSview and SMOS-Box).

SMOSview is a JAVA based Graphical User Interface (GUI) software, that decodes SMOS products, displays their content graphically or numerically as tables and exports the data to alternative formats. SMOSview is not intended for a detailed analysis, visualization or processing of SMOS data but only for quick and easy look at the raw data products.

Similarly SMOS-Box is a module for the BEAM GUI software dedicated to the decoding, visualization and basic analysis of all operational data products (Levels 1 and 2) from the SMOS mission (see SMOS/Toolbox+BEAM website). SMOS-Box imports SMOS L1C and L2 science data products into the BEAM software and use the existing BEAM functionality to decode, display, manipulate, analyze and export SMOS data products to alternative formats in an interactive way or as batch processing. The BEAM software development has, however, been discontinued.

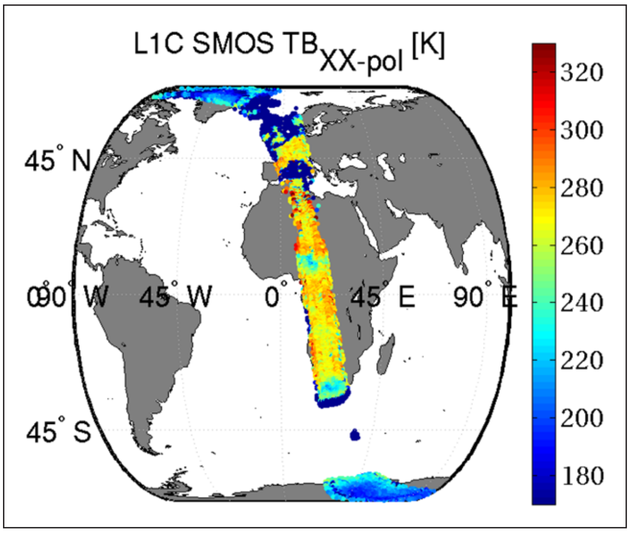

Figure 1: SMOS unprocessed brightness temperatures at X polarization from a morning overpass half-orbit L1C data file for 2 July, 2015. The color scale has been limited to 330 K, but the data contains values as high as 900 K indicating pixels with RFI.

The Sentinel Application Platform (SNAP), the successor of BEAM, is a software developed as a common architecture for all Sentinel toolboxes (i.e. Sentinel-1 to -4 and SMOS, see http://step.esa.int/main/toolboxes/snap/). SNAP utilizes the SMOS-Box toolbox and allows to convert any SMOS data levels to NetCDF using its graph processing command line tool or a graphic user interface, also for specific regions of interest. SNAP also provides the Python module `snappy`, which allows the access to its SNAP Java API from Python; hence it can also work in batch processing mode.

The *Centre Aval de Traitement des Donnes* provides a Level 3 (L3) brightness temperature product in daily archives, which comprise daily global-coverage products that include all brightness temperatures in full polarization and transformed from the antenna frame (L1B) to the Earth's surface polarization reference frame. The brightness temperatures (*TB*s) are binned and averaged for fixed angle classes. L3 *TB*s are already re-sampled on an Equal-Area Scalable Earth-grid (EASE-grid) [1] with 25 km resolution on a global scale; no detailed processing can be done in regions of interests at higher resolution on the L1C data product on the 15 km ISEA-grid.

Motivation

The discussed software tools mostly work with SMOS data in an interactive way with single data files at a time or via batch jobs. They load the whole data file and the user manually interacts with the software to glance through specific snapshots or regions for a given date. These tools are very helpful for a case-by-case analysis, but when a specific regions of interest need to be analyzed for a longer time period and large amounts of files must be processed limitation in term of reading and processing time becomes obvious. Since BEAM and SNAP are Java based they need to be interpreted by a Java Virtual Machine, which slows down its execution. The `snappy` module is a wrapper, which calls SNAP from Python and its performance, in terms of speed, might be even lower as expected from a scripting language. Therefore when many SMOS data files need to be fed to observation operators, e.g. for inversion algorithms or for data assimilation, speed-ups for reading and processing are paramount.

We developed the "`Process_SMOSxL1C`" mex-function as an alternative tool for rapid reading and processing of SMOS L1C data products with Matlab® [9] or GNU/Octave [4] programming languages. A mex-function is a C/C++ source code compiled in order to run within the Matlab workspace as a native Matlab function at a high execution speed. The "`Process_SMOSxL1C`" mex-function gives the user the ability to either work with a single file and exploit the powerful features of Matlab and Octave for analysis and visualization, or to run batch-jobs for many files and perform the same processing for a selected region and have the results as well as the original data automatically stored as MAT-files for future analysis. In addition it is possible to easily port the function to high performance computer (HPC) systems since most of them have C/C++ as default programming languages.

Implementation and architecture

The "`Process_SMOSxL1C`" function is implemented in a sequential structure. Since the SMOS L1C datablock (DBL) is a binary file, the data needs to be loaded sequentially and sorted to different variable types according to their specification [11]. The scientific L1C datablock consists of two datasets: one contains a list of snapshots and associated information in the swath (hereafter Swath Snapshot Information or SSI) and a second one contains the list of brightness temperature (*TB*) samples and associated geophysical information for each grid point sensed by the instrument (hereafter TSF for Temperatures Swath Full). In order to process the *TB*s, information from both datasets are needed, therefore SSI and TSF needs to be loaded sequentially before a region of interest is selected.

Every piece of information contained in a L1C datablock is sorted as a Matlab structure in their corresponding dataset of origin, either in SSI or TSF. Then the "`Process_SMOSxL1C`" mex-function returns the TSF structure with the original data contained in the DBL file and also gives the possibility to optionally return the SSI structure (in case of further re-processing).

The processing procedure follows five steps:

1. data filtering according to the RFI flags and other criteria,
2. interpolation of TB_{XY} values to obtain the four components of the Stokes vector at every time step,
3. transformation from the antenna reference frame (X- & Y-polarization) to the Earth's surface reference frame (H- & V-polarization),
4. for every grid point the *TB*s are averaged for equally-spaced classes of incidence angles into 1° bins from 0 to 60°,
5. results are returned either to the Matlab workspace or saved as a MAT-file containing only the selected region of interest.

Step 1 uses the RFI flags contained in the TSF datablock for every grid point [6]. Additionally, other standard RFI detection criteria have been implemented as described by [6] and [15], namely:

$$ST_4 < 50\,K \tag{1}$$

$$\left(TB_{S1} - \overline{TB_{S1}}\right) > 5 + 4\,DTB_X \tag{2}$$

with ST_4 the 4th Stokes parameter, and $\overline{TB_{S1}} = 0.5\,(TB_X + TB_Y)$ the mean value of the halved 1st Stokes parameter for all observations in one pixel [6], and DTB_X the radiometric uncertainty for TB_X. Additional criteria (not strictly related to RFI) constrain out-of-range *TB* values by the requirement $50\,K < \sqrt{TB_X{}^2 + TB_Y{}^2} < 500\,K$. Here the upper extreme may be related to RFI.

For the step 2, the *TB* values come in sequence for X-pol, Y-pol or XY, YX-pol at every snapshot time step. In order to have *TB* values with four Stokes parameters, an interpolation is needed in such a way that at every time step the *TB*s contain the [X, Y, Re{XY}, Im{XY}] Stokes parameters.

Step 3 transforms the polarization reference frame via a transformation matrix given by [6] as:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} cos^2(\alpha) & sin^2(\alpha) & -cos(\alpha)sin(\alpha) & 0 \\ sin^2(\alpha) & cos^2(\alpha) & cos(\alpha)sin(\alpha) & 0 \\ sin(2\alpha) & -sin(2\alpha) & cos(2\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} TB_H \\ TB_V \\ ST_3 \\ ST_4 \end{bmatrix} \tag{3}$$

where the leftmost vector is $\mathbf{A} = [Re\{TB_X\}, Re\{TB_Y\}, 2Re\{TB_{XY}\}, -2Im\{TB_{XY}\}]^T$ for full polarization and $\mathbf{A} = [TB_X, TB_Y]^T$ for dual-polarization. The angle $\alpha = \omega_{Fa} + \alpha_{ge}$ is the sum of ω_{Fa}, the Faraday angle, and α_{ge}, the geometric rotation angle. Every matrix and vector element depends on the incidence angle of observation.

In order to obtain the Earth's polarization reference frame vector, the above matrix system is solved for $[TB_H, TB_V, ST_3, ST_4]^T$ at every incidence angle and for every pixel-point by using the GNU Scientific Library [7].

Step 4 re-samples the vector $[TB_H, TB_V, ST_3, ST_4]^T$ at fixed incidence angle classes of 1° between 0 and 60° in the current version. This can be changed to other ranges and bin sizes, but requires re-compilation of the source code. The new re-sampled Stokes (or dual-pol) *TB*s are sorted in the following elements of the TSF structure:

```
> TSF.TB_Fixed_IncAngle,
> TSF.Fixed_IncAngle
```

where the first is an 3D array with the number of DGG pixel points in the first dimension (depending on the geographic region selected), the number of incidence angles (61 elements) in the second dimension and the polarization (either 4 or 2 elements) in the third dimension.

All information contained in the L1C datablock DBL is kept inside the structures TSF and SSI, with TSF having additional information related to the specific processing; therefore any step can be repeated or re-calculated within Matlab/Octave workspace. For a detailed description of the complete list of elements in the TSF and SSI structures, see the mex-function documentation at https://github.com/pablosaa/Process_SMOS_L1C/wiki.

The general usage of the "Process_SMOSxL1C" mex-function for a desired geographic region needs to be called from the Matlab or Octave workspace as:

```
> [TSF, SSI] = Process_SMOSxL1C
               (‘DBL_filename’,...
[lat_min, lon_min, lat_max, lon_max],...
  ‘PATH_TO_OUTPUT’);
```

with 'DBL_filename' a string with the absolute location of the L1C DBL data file, the second argument the limits of a latitude and longitude box for a geographic region of interest, and the last argument an optional string which indicates where the MAT-file containing the TSF and SSI variables will be saved. At least one output argument is mandatory, i.e. TSF, while SSI is optional.

Figure 2 shows the TB_H extracted and processed from the raw data file shown in **Figure 1** for the Neckar catchment in south-west Germany, with the limits of the geographic region given by [47.5, 7.5, 50, 10] and assigned to the second argument of "Process_SMOSxL1C".

See **Table 1** for a reference benchmark regarding the processing time for a full archive with half-orbits as in **Figure 1** and for a selected region as in **Figure 2**.

Quality control

The "Process_SMOSxL1C.cpp" code has been successfully compiled with GNU C++ version 4.7.1 in a Linux 64bit system. The compiled mex-function has been tested in three different Matlab versions: v7.8.0.347 (R2009a), v7.11.0 (R2010b), and v9.1.0.441655 (R2016b). Similarly, the code has been compiled and tested for Octave version 3.6.2. and 4.0.3.

The processing speed is listed in **Table 1** for two case studies: a whole SMOS L1C data product containing one half-orbit (see **Figure 1**), and for a region of interest limited by its bottom-left corner [47.5°N, 7.5°E] and upper-right corner [50°N, 10°E] latitude and longitude, respectively (see **Figure 2**). This region contains the Neckar river catchment and is used for data assimilation studies by [12, 14] using of the "Process_SMOSxL1C" mex-function tool.

Table 1 includes in its 2nd column the number of Digital Global Grid pixel points contained. The 3rd column shows the result of only loading the raw L1C data product into the workspace by a Matlab m-script without using the mex-function, and the 4th and 5th columns indicate the total processing time by using the "Process_SMOSxL1C" mex-function. The last row in **Table 1** indicates the percentage of minimum time needed in addition, when the option to save the processing outputs as a MAT-file is used.

Table 1 clearly shows the efficiency in terms of processing time by the mex-function, with Octave being slightly faster than Matlab. When the processed data is saved as MAT-file, Octave requires more time because all variables need to be translated to the MATIO library variable types [8].

For validation purposes the processed SMOS observations are "compared" against measurements by NASA's Soil Moisture Active Passive (SMAP) mission launched in March 2015. SMAP carries a radiometer sensing microwave radiation at 1.4 GHz at one fixed incidence angle of 40° [5, 2]. SMAP has an antenna configuration different from SMOS, and SMAP's L1C data products (36 km resolution) are delivered with the Earth's polarization reference frame, i.e. no further processing is needed.

For this validation exercise the region corresponding to the Neckar catchment has been selected. Observations by SMOS and SMAP from April to December 2015 have been taken into account when both satellites have a morning

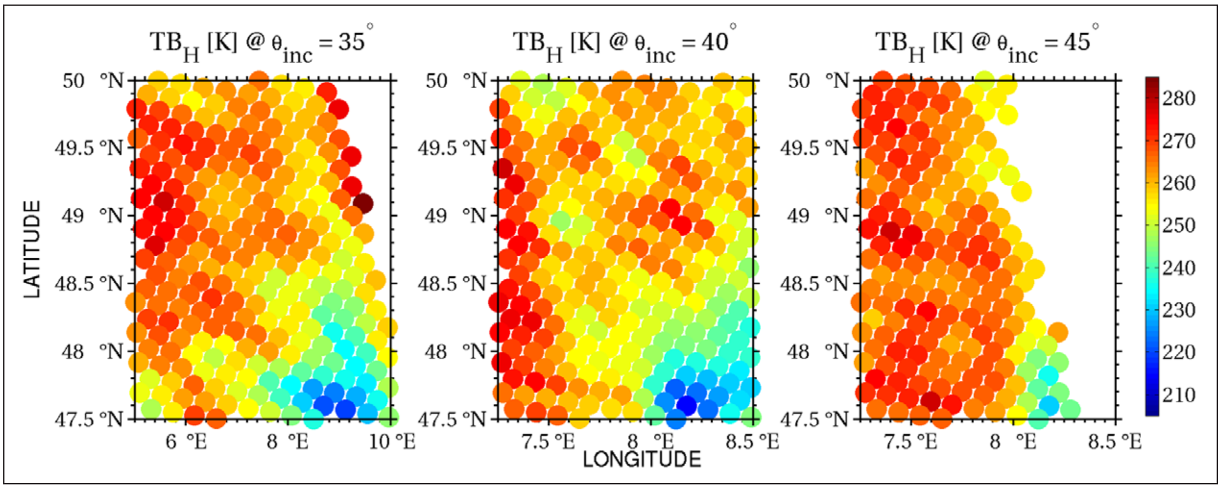
TB_H [K] @ $\theta_{inc} = 35^\circ$
TB_H [K] @ $\theta_{inc} = 40^\circ$
TB_H [K] @ $\theta_{inc} = 45^\circ$
LATITUDE
50 °N
49.5 °N
49 °N
48.5 °N
48 °N
47.5 °N
6 °E
8 °E
10 °E
7.5 °E
8 °E
8.5 °E
LONGITUDE
280
270
260
250
240
230
220
210

Figure 2: Brightness Temperature extracted for the Neckar catchment after precessing for three incidence angles 35° (left), 40° (middle) and 45° (right).

Table 1: Benchmarks for processing time with Matlab and Octave for two case studies: Half-orbit and a region (Neckar catchment). Linux 64bit PC with 4 GB RAM and CPU 2534 MHz Inter Centrino 2 core. The "m-script" column represents only loading the raw data, without any processing.

		Processing Time in Seconds		
Case Study	**No. DGGs**	**Matlab m-script**	**Matlab MEX**	**Octave MEX**
Half-orbit	114369	2957.16	84.01	82.73
Sub-sector	265	–	6.12	6.02
Storage as output MAT-file		+3%	+25%	+33%

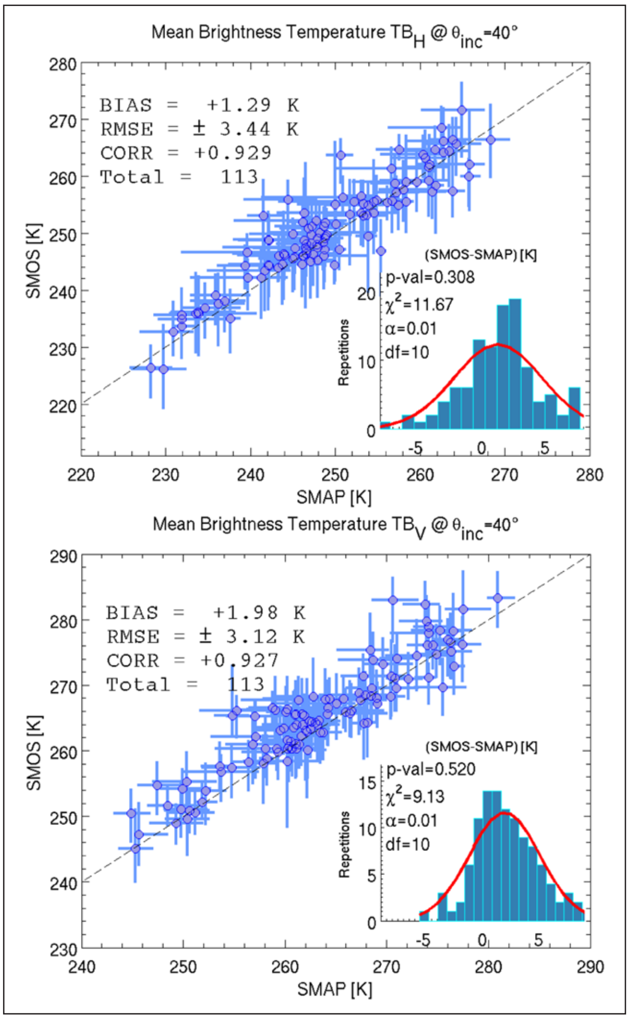

Figure 3: Statistics for SMOS vs SMAP brightness temperatures at H-polarization (top panel) and V-polarization (bottom panel) for April to December 2015. SMOS is the result of the "Process_SMOSxL1C" processing software. The dots represent the median *TB*s within the selected region and the bars are the 1st and 3rd quartiles.

overpass at the same day. **Figure 3** shows the statistics of *TB*s at H-, and V-polarization after SMOS *TB* has been processed with the "Process_SMOSxL1C" mex-function. Note the good agreement despite the fact that the data come from two completely different instruments and methodologies.

Figure 4 shows the excellent agreement in dynamic range and daily variability between SMOS and SMAP *TB*s after processing the SMOS L1C product, which lends strong confidence to the capabilities of the software to adequately yield results free of artifacts which could jeopardize further applications of the data.

Compilation and testing

The best way to ensure the that "Process_SMOSxL1C" works for a particular system is by compiling the C++ source code on the used system. To that goal, the user needs first to assure, that Matlab's mex compiler is properly configured and tested. Compilation procedures are indicated in the https://github.com/pablosaa/Process_SMOS_L1C/wiki. The user can also try the pre-compiled version provided and test whether or not the Process_SMOSxL1C mex-function is working. The easiest way to test is by trying to process a single file interactively from the Matlab or Octave workspace, following the simple command:

```
> TSF = Process_SMOSxL1C;
```

Immediately, a file browser pops-up for navigating thought directories to select a SMOS L1C DBL file. Once selected another dialogue-box pops-up (for the Matlab version) for entering the latitude and longitude limits for the processing region (these are done provided as command line inputs for the Octave version). If the Process_SMOSxL1C mex-function shows any message or warnings, it could mean that the input arguments are wrong, but indicates already that the mex-function is working in the system. When no warning appears the process should be working and the user can look at the content of the TSF structure.

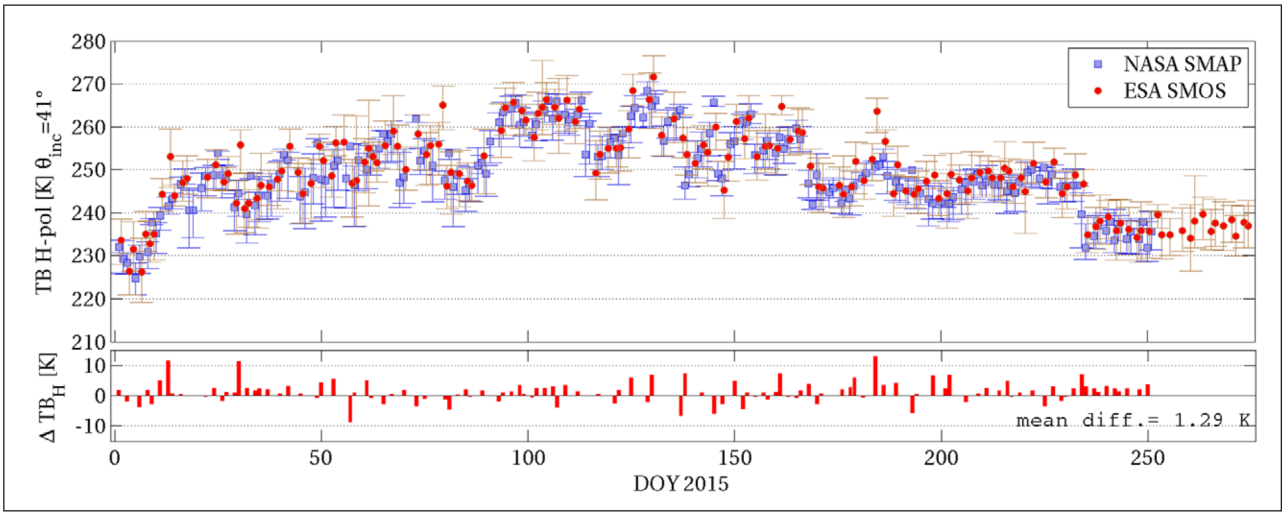

Figure 4: Time series of SMAP brightness temperature measurements and SMOS L1C product processed with "Process_SMOSxL1C" mex-function (top). The bottom panel shows the residuals of the median within the distribution.

Note that the "Process_SMOSxL1C" mex-function has as file extension either ".mex" or ".mexa64" for Octave and Matlab, respectively. "Process_SMOSxL1C" requires that the corresponding L1C DBL and HDR files are located in the same directory. Further description and examples on how to run the function and plot results can be found at https://github.com/pablosaa/Process_SMOS_L1C/wiki.

(2) Availability

Operating system

GNU/Linux kernel 3.4.63-2.44, tested on OpenSuse version 12.2 or higher.

Programming language

Source code is written in GNU C++ language.

Additional system requirements

Minimum Memory RAM: 2 GB, disk space depending on number of half orbit L1C binary data files (each file is ~435 MB).

Dependencies

To compile the software the MEX library is needed, with two possibilities for compilation:

- as a MATLAB® function via MEX compiler, revision: 1.1.6.3 [9],
- as a GNU/Octave [4] mex function via MKOCTFILE compiler version 3.6.2 or higher.

The program requires:

- The GNU Scientific Library (GSL) version 2.2.1 for C and C++ language,
- The GNU C GSL Basic Linear Algebra Subprograms (GSLCBLAS) version 3.6.0. [7],
- MATIO library version 1.5.9 [8] is required for Octave to support MAT-file archiving.

List of contributors

Pablo Saavedra Garfias: project scientist, Meteorological Institute, University of Bonn.

Clemens Simmer: project Principal Investigator (PI), Meteorological Institute, University of Bonn.

Software location

Archive

Name: Process_SMOS_L1C-v1.0.zip
Persistent identifier: https://doi.org/10.5281/zenodo.1036856
Licence: GNU General Public License version 3
Publisher: Pablo Saavedra Garfias
Version published: v1.0
Date published: 26/10/2017

Code repository

Name: GitHub
Persistent identifier: https://github.com/pablosaa/Process_SMOS_L1C
Licence: GNU General Public License version 3
Date published: 13/02/2017

Language

English

(3) Reuse potential

SMOS datasets are widely used for research and application in studies regarding ocean salinity, soil moisture, root zone soil moisture, ice-thickness, landslides monitoring,

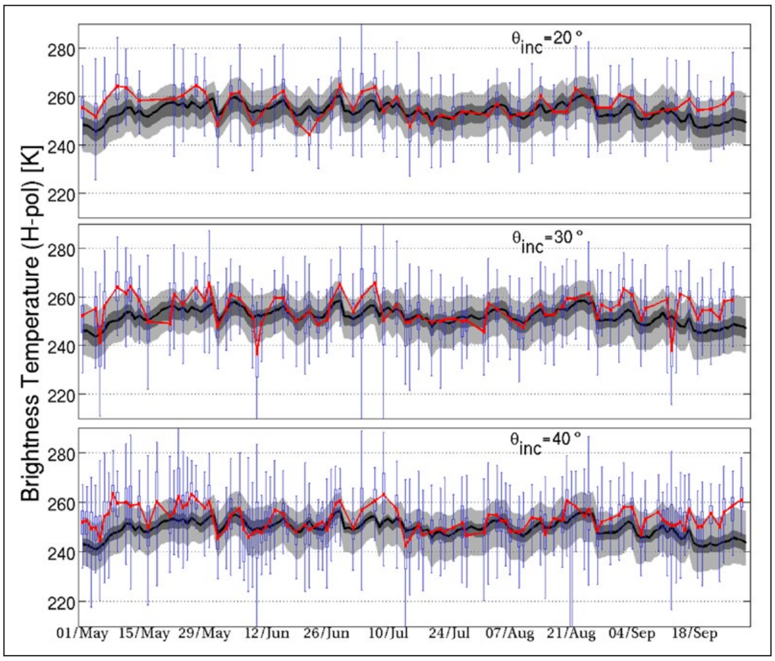

Figure 5: Simulation of brightness temperatures at 1.4 GHz and three incidence angles: 20° (top), 30° (center) and 40° (bottom). The black-line is the median of the *TB* distribution within the Neckar catchment, the gray area is the inter-quantile region and the lighter gray area indicate the 5% to 95% quantiles. The Box-plots depict the statistics of the distributions of the real measurements by SMOS and processed by "Process_SMOSxL1C" mex-function.

and precipitation estimation, among others. The "Process_SMOSxL1C" mex-function can be used by researchers, who need to work directly with brightness temperatures rather than with other geophysical parameter retrievals provided in L2 or L3 products. It can also be used to develop or test other retrieval approaches. With the SMOS satellite in operation since 2009, time series are becoming rich in terms of information of the Earth's surface and ocean sensed by the SMOS instrument. A rapid tool to process and re-analyze large numbers of data files is becoming a necessity and "Process_SMOSxL1C" can be used to that goal.

The "Process_SMOSxL1C" mex-function is being used for calibration and comparisons with synthetic observations obtained by satellite simulators alike SMOS or SMAP by [12]. The calibrated synthetic brightness temperatures are then utilized as state variables for data assimilation experiments applied to land-surface-atmosphere coupled models as shown by [14]. For instance, **Figure 5** depicts the agreement of simulations by the Community Microwave Emission Model (CMEM) [3] after being calibrated against SMOS multi-incidence angle brightness temperatures processed by "Process_SMOSxL1C" mex-function. The box-plots are composed by the median (read dot), the inter-quantile region (blue box), the extremes (whiskers), and outliers as green-dots.

The "Process_SMOSxL1C" mex-function is documented online at https://github.com/pablosaa/Process_SMOS_L1C/wiki.

So far the "Process_SMOSxL1C" mex-function is able to work only with L1C scientific data files (SC), however SMOS also provides the L1C browse product (BW) which only contains data for the incidence angle of 42.5°. "Process_SMOSxL1C" is not yet adjusted to support this browse product, but any contribution is welcome by making a pull request to the Process_SMOS_L1C GitHub repository. Any support mechanisms is based on a voluntary basis via the repository.

Acknowledgements

The authors thank to the German Science Foundation (*Deutsche Forschungsgemeinschaft* DFG) for funding the research unit FOR2131 and its sub-project P2 on "Scale-Problems in Assimilation of Passive Microwave Observation into Coupled Models". Special thanks to the research group FOR2131, in particular to Dr. Bernd Schalge, for providing the Community Land Model (CLM) outputs to perform the SMOS simulations in **Figure 5**.

Competing Interests

The authors have no competing interests to declare.

References

1. **Brodzik, M J** and **Knowles, K W** 2002 *EASE-Grid: A Versatile Set of Equal-Area Projections and Grids*. In: Goodchild, M (ed.), Discrete Global Grids. Santa Barbara, California USA: National Center for Geographic Information & Analysis.
2. **Chan, S, Njoku, E** and **Colliander, A** 2014 *Soil Moisture Active Passive (SMAP), Algorithm Theoretical Basis Document Level 1C Radiometer Data Product*. Jet Propulsion Laboratory.
3. **de Rosnay, P, Drusch, M, Boone, A, Balsamo, G, Decharme, B, Harris, P, Kerr, Y, Pellarin, T, Polcher, J** and **Wigneron, J P** 2009 Amma land surface model intercomparison experiment coupled to the comunity microwave emission model: Almip-mem. *J. Geophys. Res.*, 114(D05): 108.
4. **Eaton, J W, Bateman, D, Hauberg, S** and **Wehbring, R** 2014 *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform. ISBN: 1441413006.
5. **Entekhabi, D** and **Coauthors** 2010 The soil moisture active and passive (SMAP) mission. *Proc. IEEE*, 98: 704–716. DOI: https://doi.org/10.1109/JPROC.2010.2043918
6. **ESA** 2011 Algorithm theoretical basis document (ATBD) for the SMOS L2 soil moisture processor development continuation project. Technical report, European Space Agency. ESA No.: SO-TN-ARR-L2PP-0037.
7. **Galassi, M, Davies, J, Theiler, J, Gough, B, Jungman, G, Alken, P, Booth, M, Rossi, F** and **Ulerich, R** 2015 *GNU Scientific Library Reference Manual*. http://www.gnu.org/software/gsl/, 3rd edition. ISBN: 0954612078.
8. **Hulbert, C C** 2016 *MATIO User manual for version 1.5.9*. https://matio.sourceforge.io/.
9. **MathWokrs Inc** 2010 *MATLAB version 7.10.0 (R2010b)*. The MathWorks Inc., Natick, Massachusetts.
10. **Kerr, Y H, Waldteufel, P, Wigneron, J P, Front, J** and **Berger, M** 2001 Soil moisture retrieval from space: The soil moisture and ocean salinity (SMOS) mission. *IEEE Trans. Geosci. Re. Sens.*, 39(8): 1729–1735. DOI: https://doi.org/10.1109/36.942551
11. **Indra, S A** 2015 SMOS DPGS: SMOS level 1 and auxiliary data products specifications. Technical report, INDRA sistemas S.A., Ref. SO-TN-IDR-GS-0005.
12. **Saavedra, P, Simmer, C** and **Schalge, B** 2016 Evaluation of modeled high resolution virtual brightness temperatures compared to space-borne observations for the neckar catchment. In: *14th Specialist Meeting on Microwave Radiometry and Remote Sensing of the Environment*, 85–90. IEEE. DOI: https://doi.org/10.1109/MICRORAD.2016.7530510
13. **Sahr, K, White, D** and **Kimerling, A J** 2003 Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2): 121–134. DOI: https://doi.org/10.1559/152304003100011090
14. **Schalge, B, Rihani, J, Baroni, G, Erdal, D, Geppert, G, Haefliger, V, Haese, B, Saavedra, P, Neuweiler, I, Hendricks, H J, Ament, F, Attinger, S, Cirpka, O A, Kollet, S, Kunstmann, H, Vereecken, H** and **Simmer, C** 2017 High-resolution virtual catchment simulations of the subsurface-land surface-atmosphere system. *Discussion paper*. In preparation.
15. **Schlenz, F, dall Amico, J T, Mauser, W** and **Loew, A** 2012 Analysis of smos brightness temperature and vegetation optical depth data with coupled land surface and radiative transfer models in southern germany. *Hydrology and Earth System Sciences*, 16: 3517–3533. DOI: https://doi.org/10.5194/hess-16-3517-2012

How to cite this article: Saavedra, P and Simmer, C 2017 An Octave/MATLAB® Interface for Rapid Processing of SMOS L1C Full Polarization Brightness Temperature. *Journal of Open Research Software,* 6: 2. DOI: https://doi.org/10.5334/jors.165

Submitted: 16 February 2017 **Accepted:** 03 November 2017 **Published:** 08 January 2018

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS