Google Earth Engine tool to generate soil moisture maps with Sentinel-1 satellite imagery

Authors: Harm-Jan Benninga (h.f.benninga@utwente.nl), Michiel Pezij

Date: 4 March 2019

1. Introduction

This document describes the tool that we developed in the Google Earth Engine to generate soil moisture maps from Sentinel-1 satellite imagery. This work was done within the research programme OWAS1S (Optimizing Water Availability with Sentinel-1 Satellites). Information about the project can be found on www.owas1s.nl.

In this document we explain the basics behind the algorithm and refer to other sources for more information. In Section 2 we describe the satellite imagery that is used, in Section 3 we describe the code in the Google Earth Engine and in Section 4 we describe the availability of the code and necessary input maps.

2. Satellite data

The tool uses Sentinel-1 imagery. These images are collected by the Sentinel-1A and Sentinel-1B satellites, which are operated by the European Space Agency (ESA) as part of the Copernicus Programme. Sentinel-1A provides images since 3 October 2014 and Sentinel-1B since 26 September 2016. The Sentinel-1 images that are provided in the Google Earth Engine have already been pre-processed, as described on https://developers.google.com/earth-engine/sentinel1.

We used the images with the following specifications:

- Instrument mode Interferometric Wide swath (IW): this is the primary operation mode over land, so most images are available in this mode.
- Polarization VV: over land observations are primarily acquired in VV and VH polarization. Observations in VV are more sensitive to soil moisture than observations in VH. Also, less prone to vegetation impacts on scattering coefficients.
- Resolution 10 m × 10 m: the highest resolution available.
- Ascending and descending orbits: the Sentinel-1 satellites overpass the Netherlands in ascending (satellite travels northwards) and descending orbits (satellite travels southwards). We combined the ascending and descending orbits.

For detailed information about Sentinel-1 we refer to the mission guide, user guide and technical guide on https://sentinel.esa.int/web/sentinel/user-guides and the scientific paper by Torres et al. (2012).

Sentinel-1's acquisition strategy results in a temporal resolution of 1.5 days to 4 days over Europe (Bauer-Marschallinger et al., 2019; Torres et al., 2012). Figure 1 shows the temporal resolution (average number of days between two images) of Sentinel-1A and Sentinel-1B observations for the Netherlands. The temporal resolution over the Netherlands ranges from 1.5 days to 3 days.

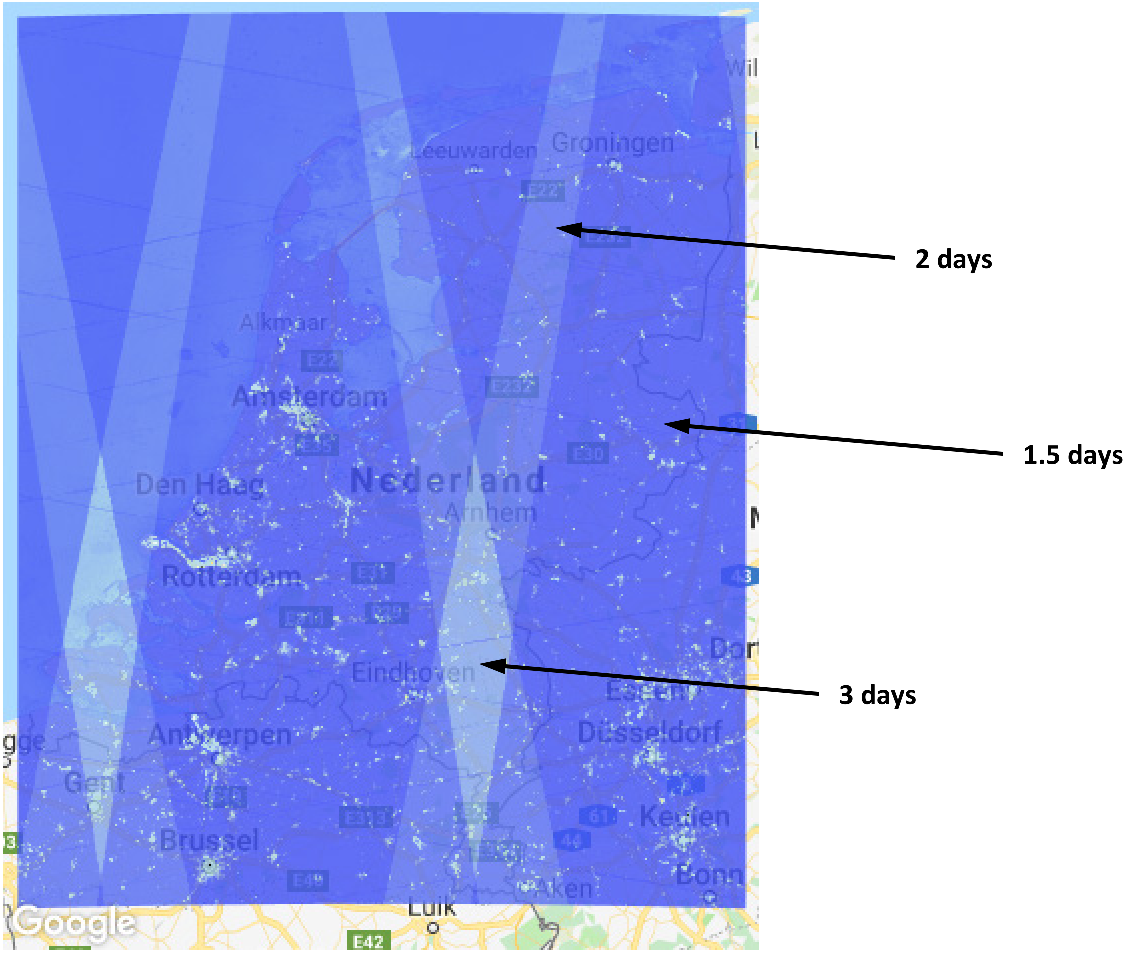

Figure 1: Average temporal resolution of Sentinel-1A and Sentinel-1B over the Netherlands (Blue: 0 days, Light blue: 4 days). Urban areas have a coarse temporal resolution (light blue colour), because for such areas the backscatter observations are generally outside the range of observation values that may carry information about soil moisture and for that reason these observations were filtered out (see Section 3.5).

3. Google Earth Engine code

In this section we explain the function and background of each part of the code. The web-based code editor of Google Earth Engine uses the programming language JavaScript. For detailed information about Google Earth Engine, we refer to Gorelick et al. (2017) and the user manual on https://developers.google.com/earth-engine/.

3.1 Import section

In this first section, before the actual code, additional data, like maps of soil moisture content at the wilting point and saturated conditions, and geometries are imported. Other study areas can be drawn using 'Geometry Imports' in the mapping section in Google Earth Engine (new layers are automatically added to the code).

```
var Netherlands = /* color: #ffc82d */ee.Geometry.Polygon(
        [[[3.3, 53.7],
          [3.3, 50.7],
          [7.3, 50.7],
          [7.3, 53.7]]]),
    Noordoostpolder = /* color: #98ff00 */ee.Geometry.Polygon(
        [[[5.787788433740957, 52.66411371687477],
          [5.863319439600332, 52.59346991172528],
          [5.961166424463613, 52.635161466372686],
```

```
                    [5.889068646143301, 52.69762438093982],
                    [5.886925002732482, 52.69992440679381]]]),
        SAT_5cm = ee.Image("users/hjbenninga/BOFEK2_1_VGE_SAT_5cm"),
        WP_5cm = ee.Image("users/hjbenninga/BOFEK2_1_VGE_WP_5cm");
```

3.2 Basic settings

Some basic settings are defined here, also see the comments behind the settings in green font colour. `Area` is the total spatial domain over which Sentinel-1 images are included in the calculations. `Area_small` is the domain for which outputs are generated.

```
// Authors: Harm-Jan Benninga & Michiel Pezij
// Date: 26 February 2019


// --- Instructions ---
//  - For the value of a pixel, go to Inspector (tab on right bar) and click on the map viewer
//  - Turn off/on maps in the viewer (below) at 'Layers' (right)
//  - To make a soil moisture map for another area:
//       1. Draw a new polygon in the viewer (below) at 'Geometry Imports' (left)
//       2. Define 'Area_small' (line 16) as this new polygon

// --- Settings ---
var EXPORT_image = false;    // Make 'true' to export an image for 'Date_image' for 'Area_small'. Then go to 'Tasks' (tab on right bar) to start the export to Google Drive.
var EXPORT_table = false;     // Make 'true' to export a table of average soil moisture over time for 'Area_small'
var MOSAIC = false;           // Make 'true' to mosaic multiple images that cover Area_small on Date_image

var SCALE = 100;              // Scale [meter] of exported image and table

var Area = Netherlands;
var Area_small = Noordoostpolder_polder;            // Change this to plot for another area
var Date_image = ee.Date('2019-01-07T00:00:00');    // Change this date to plot and export image for another date


```

3.3 Loading of Sentinel-1

The algorithm to retrieve the soil moisture maps is based on statistics over a time series. The Sentinel-1 time series were selected based on the specifications listed in Section 2. The complete set of images (all images available) is included in an image collection. To calculate the time series statistics we used three complete hydrological years (1 March 2016 to 1 March 2019).

```
// --- Load Sentinel-1 C-band SAR Ground Range collection (log scaling, VV co-polar) ---
var collection_S1_TOTAL = ee.ImageCollection('COPERNICUS/S1_GRD').filterBounds(Area)
.filter(ee.Filter.listContains('transmitterReceiverPolarisation','VV'))
  .filter(ee.Filter.eq('instrumentMode','IW'))
  .filterDate('2014-10-01', Date.now())
```

```
29.   .filter(ee.Filter.eq('resolution_meters',10));
30.
31. var collection_S1_STATS = collection_S1_TOTAL.filterDate('2016-03-01',
    '2019-03-01');    // Statistics are calculated over three complete
    hydrological years
32.
33.
```

3.4 Incidence angle correction

The incidence angle under which Sentinel-1 observes the earth surface depends on the location and the orbit in which the Sentinel-1 image is acquired. The incidence angle can vary between 29.1° and 46°. We corrected the Sentinel-1 observations to an incidence angle of 37.5° by a cosine correction. The coefficient `n` is equal to 1 if reradiation of the satellite signal from the earth surface is isotropic (volume scattering) and `n` is equal to 2 if reradiation follows Lambert's cosine law (Ulaby et al., 1982; Van der Velde et al., 2015). Here we assume `n = 2`, because previous studies obtained good results with this assumption (Lievens et al., 2011; Mladenova et al., 2013; Van der Velde and Su, 2009). This is, however, a rough assumption because in reality the type of reradiation may depend on surface characteristics such as the vegetation cover.

```
34. // --- Incidence angle correction ---
35. var n = 2;              // normalization coefficient
36. var angle_ref = 37.5;    // reference angle
37.
```

The function to normalize the Sentinel-1 observations to `angle_ref` is defined in Line 38 – Line 60. This function contains three operations:

- Line 40 – Line 43: The Sentinel-1 observations are provided in decibel scale, but the cosine correction requires values in the nonlogarithmic domain:

$$[dB] = 10 * \log_{10}\left(\left[\frac{m^2}{m^2}\right]\right) \rightarrow \left[\frac{m^2}{m^2}\right] = 10^{\frac{[dB]}{10}}$$

  Only the observations in the VV band are selected, as explained in Section 2.

- Line 45 – Line 52: This is the actual cosine correction function. Each Sentinel-1 image contains local incidence angles in the band `'angle'`.

- Line 54 – Line 57: After the cosine correction the Sentinel-1 observations are converted back to the decibel scale.

Finally, time stamps are copied from the Sentinel-1 images and attached to the new images (Line 59).

```
38. // define incidence angle correction function
39. var incidence_angle_correction_function = function(image) {
40.   var image_m2m2 = image.expression(
41.     '10**(image/10)', {
42.       'image': image.select('VV')
43.     });
44.
45.   var image_m2m2_cor = image_m2m2.expression(
46.     'sigma0*((cos(pi/180*angle_ref)**n)/(cos(pi/180*angle)**n))', {
47.       'n': n,
48.       'angle_ref': angle_ref,
49.       'sigma0': image_m2m2,
50.       'angle': image.select('angle'),
51.       'pi': Math.PI
52.     });
53.
```

```
54.   var output_image = image_m2m2_cor.expression(
55.     'log10(sigma0_cor)*10', {
56.       'sigma0_cor': image_m2m2_cor
57.     });
58.
59.   return output_image.set('system:time_start',
    image.get('system:time_start'));
60. };
61.
```

Line 38 – Line 60 defines the incidence angle correction function. In Line 63 and Line 64 the function is applied to all images in the image collections `collection_S1_STATS` and `collection_S1_TOTAL`, respectively.

```
62. // apply incidence angle correction function
63. var collection_S1_STATS_IC_cor =
    collection_S1_STATS.map(incidence_angle_correction_function);
64. var collection_S1_TOTAL_IC_cor =
    collection_S1_TOTAL.map(incidence_angle_correction_function);
65.
```

3.5 Mask Sentinel-1: valid observation values

Very high and very low backscatter observations are filtered, because these are unlikely to carry information about soil moisture (Bauer-Marschallinger et al., 2019). Unrealistic values below -20 dB occur on the boundaries of Sentinel-1 images. To avoid artefacts on the boundaries of Sentinel-1 images, we set the lower limit of valid values at -20 dB.

All Sentinel-1 observations below `min_value` and above `max_value` are filtered.

```
66. // --- Mask Sentinel-1 observations ---
67. var min_value = -20;           // Minimum value that is valid
68. var max_value = -2;            // Maximum value that is valid
69. var min_coverage_ratio = 0.75  // Minimum ratio of images with a value
    between min_value and max_value
70.
71. // Define masking function
72. var valid_values_mask_function = function(image) {
73.   var lower_mask = image.gte(min_value);
74.   var new_img = image.updateMask(lower_mask);
75.
76.   var upper_mask = new_img.lte(max_value);
77.   var new_img2 = new_img.updateMask(upper_mask);
78.
79.   return new_img2.set('system:time_start',
    image.get('system:time_start'));
80. };
81.
82. // Apply threshold values to image collections
83. var collection_S1_STATS_IC_cor_Masked =
    collection_S1_STATS_IC_cor.map(valid_values_mask_function);
84. var collection_S1_TOTAL_IC_cor_Masked =
    collection_S1_TOTAL_IC_cor.map(valid_values_mask_function);
85.
```

3.6 Mask Sentinel-1: coverage of a pixel

Pixels that contain information about soil moisture will generally have Sentinel-1 observations between `min_value` and `max_value`. We mask pixels for which over the time series a ratio lower than `min_coverage_ratio` has a valid observation value. These pixels are typically built-up or water areas.

```
86. // Determine ratio of images with valid values
87. var count_collection_TOTAL = collection_S1_TOTAL_IC_cor.reduce(ee.Reducer.count());
88. var count_collection_TOTAL_masked = collection_S1_TOTAL_IC_cor_Masked.reduce(ee.Reducer.count());
89. var count_collection_STATS = collection_S1_STATS_IC_cor.reduce(ee.Reducer.count());
90. var count_collection_STATS_masked = collection_S1_STATS_IC_cor_Masked.reduce(ee.Reducer.count());
91.
92. var valid_coverage_STATS_ratio = count_collection_STATS_masked.divide(count_collection_STATS);
93. var valid_coverage_TOTAL_ratio = count_collection_TOTAL_masked.divide(count_collection_TOTAL);
94.
95. // Apply filter of min_coverage_ratio
96. var collection_S1_STATS_IC_cor_Masked2 = collection_S1_STATS_IC_cor_Masked.map(function(img) {
97.                 var mask = valid_coverage_STATS_ratio.gte(min_coverage_ratio);
98.                 var new_img = img.updateMask(mask);
99.
100.                     return new_img;
101.                     });
102.
```

3.7 Change Detection statistics

We use the change detection algorithm, introduced by Wagner et al. (1999), to generate soil moisture maps. This is a linear scaling of backscatter observations between minimum and maximum observed backscatter, for each pixel. The main assumptions are that all changes in Sentinel-1 observations can be attributed to soil moisture and that the relation between changes in Sentinel-1 observations and changes in soil moisture is linear. No detailed ground information is required for this algorithm. The model is built on a statistical analysis of the Sentinel-1 time series by scaling between minimum backscatter (observed under dry conditions) and maximum backscatter (observed under wet conditions):

$$\mathrm{cd\_s1} = \frac{\mathrm{s1\_im} - \mathrm{s1\_min}}{\mathrm{s1\_max} - \mathrm{s1\_min}},$$

where `cd_s1` is a relative soil moisture index and `s1_im` is a Sentinel-1 backscatter observation. The absolute minimum and maximum Sentinel-1 backscatter observations are probably outliers. Therefore, `s1_min` is defined by the 2.5% percentile and `s1_max` by the 97.5% percentile. For more background on the change detection algorithm, we refer to Wagner et al. (1999), Bauer-Marschallinger et al. (2019) and Van der Velde et al. (2018). `s1_min` and `s1_max` are determined for the Sentinel-1 image collection containing full hydrological years (loaded in Line 31), see Line 107 and Line 108.

```
103.    // --- Get statistics as input to Change Detection ---
104.    //var max_collection = collection_S1_STATS_IC_cor_Masked2.reduce(ee.Reducer.max())     // Maximum in each pixel
105.    //var min_collection = collection_S1_STATS_IC_cor_Masked2.reduce(ee.Reducer.min())     // Minimum in each pixel
106.
```

```
107.    var max_collection = collection_S1_STATS_IC_cor_Masked2.reduce(ee.Reducer.percentile([97.5]));   // 97.5% percentile to exclude outliers
108.    var min_collection = collection_S1_STATS_IC_cor_Masked2.reduce(ee.Reducer.percentile([2.5]));    // 2.5% percentile to exclude outliers
109.
110.
```

3.8 Retrieve a relative soil moisture index

Line 113 – Line 121 defines the scaling function of `s1_im` between `s1_min` and `s1_max` to retrieve the relative soil moisture index `cd_s1` (the equation in Section 3.7). In Line 124 the function is applied to the image collection containing all Sentinel-1 images.

```
111.    // --- Define change detection function ---
112.
113.    var change_detection_function = function(image) {
114.      var output_image = image.expression(
115.        '(s1_im - s1_min) / (s1_max - s1_min)', {
116.          's1_im': image,
117.          's1_min': min_collection,
118.          's1_max': max_collection
119.        });
120.      return output_image.set('system:time_start', image.get('system:time_start'));
121.    };
122.
123.    // apply change detection
124.    var cd_s1 = collection_S1_TOTAL_IC_cor_Masked.map(change_detection_function);
125.
126.
```

3.9 Retrieve volumetric soil moisture

`cd_s1` indicates the relative saturation of the soil as a value between 0 and 1. We linearly scale `cd_s1` to volumetric soil moisture `cd_s1_volumetric`, by assuming that soil moisture varies between the wilting point `WP` and saturated conditions `SAT`:

$$\text{cd\_s1\_volumetric} = (\text{SAT} - \text{WP}) \times \text{cd\_s1} + \text{WP}.$$

This function is defined in Line 130 – Line 138. Line 141 applies this function to the complete image collection of relative soil moisture indices. For maps of `SAT` and `WP` we use information from the BOdemFysische EenhedenKaart version 2.1 (BOFEK2012) (from https://www.wur.nl/nl/show/Bodemfysische-Eenhedenkaart-BOFEK2012.htm, also see Wösten et al. (2013)), aggregated to a raster with a resolution of 250 m. Soil moisture at wilting point and saturated conditions is calculated using the Van Genuchten (1980) equation.

```
127.    // --- Scale between Wilting Point (WP) and Saturation (SAT) ---
128.    // Wilting point and Saturation soil moisture content are adopted from BOFEK2012.
129.
130.    var WP_SAT_scale_function = function(image) {
131.      var output_image = image.expression(
132.        '(MAX - MIN) * index + MIN', {
133.          'index': image,
134.          'MIN': WP_5cm,
135.          'MAX': SAT_5cm
136.        });
```

```
    return output_image.set('system:time_start', image.get('system:time_start'));
  };

  // apply scaling between WP and SAT
  var cd_s1_volumetric = cd_s1.map(WP_SAT_scale_function);
  print(cd_s1_volumetric,'Image collection of volumetric soil moisture maps');

```

3.10 Maps and graphs

Several maps and graphs are directly plotted (see the 'Console' tab on the right bar in the Google Earth Engine).

```
   // --- Map display ---

  // Mosaic images on same day that cover Area_small

  if(MOSAIC === true) {
    var date_object = ee.Date(ee.Image(cd_s1_volumetric.filterDate(Date_image,Date.now())
        .filterBounds(Area_small).first()).get('system:time_start'));

    var cd_s1_volumetric_date = ee.ImageCollection(cd_s1_volumetric.filterDate(date_object,date_object.advance(1,'day'))
        .filterBounds(Area_small));              // Select images that cover area of interest on/after Date_image

    var image1 = ee.Image(cd_s1_volumetric_date.first());
    var date_image1 = ee.Date(image1.get('system:time_start'));

    var cd_s1_volumetric_date_image = cd_s1_volumetric_date.mosaic().set('system:time_start', date_image1);

    var cd_s1_volumetric_date_image_Study_area = cd_s1_volumetric_date_image.clip(Area_small);   // Clip to image to area of interest

    print(cd_s1_volumetric_date_image_Study_area,'Mosaiced image covering the area of interest on date of interest');

  } else {
    var cd_s1_volumetric_date_image = ee.Image(cd_s1_volumetric.filterDate(Date_image,Date.now())
        .filterBounds(Area_small).first());              // Select first image that covers area of interest on/after Date_image

    var cd_s1_volumetric_date_image_Study_area = cd_s1_volumetric_date_image.clip(Area_small);   // Clip to image to area of interest

    print(cd_s1_volumetric_date_image_Study_area,'First image covering the area of interest on date of interest');
  }

  // Add layers to map
  Map.centerObject(Area, 6);

  Map.addLayer(SAT_5cm, {min: 0, max: 0.7, opacity:1, palette: ['ff1c05', 'fff705','4dff03','07ffe8','0501ff']},
```

```
177.      'Saturation soil moisture [m^3/m^3]');          // Add map with saturation soil moisture (from BOFEK2012)
178.    Map.addLayer(WP_5cm, {min: 0, max: 0.7, opacity:1, palette: ['ff1c05', 'fff705','4dff03','07ffe8','0501ff']},
179.      'Wilting point soil moisture [m^3/m^3]');       // Add map with wilting point soil moisture (from BOFEK2012)
180.
181.    var valid_coverage_ratio_Study_area = valid_coverage_STATS_ratio.clip(Area_small);
182.
183.    Map.addLayer(valid_coverage_ratio_Study_area, {min: 0, max: 1, opacity: 1, palette: ['LightBlue','blue']},
184.      'Ratio of valid values');
185.
186.    // map soil moisture image
187.    Map.addLayer(cd_s1_volumetric_date_image_Study_area, {min: 0, max: 0.7, opacity:1,
188.      palette: ['ff1c05', 'fff705','4dff03','07ffe8','0501ff']}, 'Volumetric soil moisture [m^3/m^3]');
189.
190.
191.    // --- Plot figure soil moisture in time ---
192.
193.    var SoilMoisture_mean_TimeSeries = ui.Chart.image.series(cd_s1_volumetric, Area_small, ee.Reducer.mean(), SCALE)
194.      .setOptions({
195.        hAxis:{title:'Date'},
196.        vAxis:{title:'Volumetric soil moisture'}
197.      });
198.
199.    print(SoilMoisture_mean_TimeSeries,'Time series of mean soil moisture for the area of interest');
200.
201.    var SoilMoisture_count_TimeSeries = ui.Chart.image.series(cd_s1_volumetric, Area_small, ee.Reducer.count(), SCALE)
202.      .setOptions({
203.        hAxis:{title:'Date'},
204.        vAxis:{title:'Number of pixels'}
205.      });
206.
207.    print(SoilMoisture_count_TimeSeries,'Time series of number of pixels for the area of interest');
208.
209.
```

3.11 Export a table and image

It is possible to export tables and images to Google Drive.

- Line 212 – Line 244: If `EXPORT_table` is set **true** (Line 14), a table with the mean volumetric soil moisture for `Area_small` and the number of pixels that cover the area can be exported. The export must be confirmed at the 'Tasks' tab on the right bar in the Google Earth Engine. It can happen that there are multiple values for one day (see point 4 in Chapter 4).

- Line 248 – Line 264: If `EXPORT_image` is set **true** (Line 13), an map containing volumetric soil moisture can be exported as a GeoTiff. The export must be confirmed at the 'Tasks' tab on the right bar in the Google Earth Engine.

However, exporting images and tables takes a considerable time. Therefore, it is recommended to do further analysis on the soil moisture maps as much as possible in the Google Earth Engine.

Time series of soil moisture estimations for a certain area could also be exported by downloading a CSV file of the time series graph (press button on the upper right of a graph). This takes much less time, probably because the calculation scale is coarser.

```
// --- Statistics in a table ---

if(EXPORT_table === true) {
  // From: https://gis.stackexchange.com/questions/274569/exporting-table-in-to-a-drive-from-google-earth-engine-returns-blank-rows

  var cd_s1_volumetric_Area_small = ee.ImageCollection(cd_s1_volumetric.filterBounds(Area_small));   // Select images that cover area of interest

  var reducers = ee.Reducer.mean().combine({
    reducer2: ee.Reducer.count(),
    sharedInputs: true
  });

  var Region_table = cd_s1_volumetric_Area_small.map(function(img) {
    return img.reduceRegions({
      collection: Area_small,
      reducer: reducers,
      scale: SCALE
    }).map(function(f){
      return f.set('Date', ee.Date(img.get('system:time_start')));
    });
  }).flatten();

  print(Region_table.limit(20),'Feature table: first # elements in time series');

  Export.table.toDrive({
    collection:Region_table,
    description: 'ResultsTable_Area_of_Interest',
    selectors:(['Date','mean','count']),
    });

  print('See tab Tasks to start exporting a table with mean and number of pixels for area of interest');
} else {
    print('No table export (see variable EXPORT_table)');
}

// --- Export the image ---

var date_object = ee.Date(cd_s1_volumetric_date_image_Study_area.get('system:time_start'));
var date_string = date_object.format('YYYYMMdd_HHmm');

print('Timestamp image: ', date_string);

if(EXPORT_image === true) {
  Export.image.toDrive({
    image: cd_s1_volumetric_date_image_Study_area,
    description: 'S1_Vol_SoilMoisture_' + date_string.getInfo(),
    scale: SCALE,           // In meter
  region: Area_small
  });

```

```
260.      print('See tab Tasks to start exporting a soil moisture map for
    area of interest');
261.    } else {
262.      print('No image export (see variable EXPORT_image)');
263.    }
```

4. Code availability and remarks

You have to apply for an account to be able to use Google Earth Engine (can take a couple of days):

https://signup.earthengine.google.com/

The Google Earth Engine code can be accessed at:

https://code.earthengine.google.com/4356ddd44767cc8e8642db472643c4d0

Maps (in the form of rasters) of soil moisture at wilting point and saturated conditions are also required. These rasters can be obtained via:

- https://code.earthengine.google.com/?asset=users/hjbenninga/BOFEK2_1_VGE_SAT_5cm
- https://code.earthengine.google.com/?asset=users/hjbenninga/BOFEK2_1_VGE_WP_5cm

After using these links, the rasters of soil moisture at wilting point and saturated conditions are placed at ‘Assets’ (tab on the left bar of Google Earth Engine). Press on ‘Import into script’ to import the rasters into your code.

The algorithm that we implemented in Google Earth Engine to generate soil moisture maps is relatively simple. The following aspects should be considered when using this tool:

1. Sentinel-1 observations are only sensitive to soil moisture for areas with a limited vegetation cover (not clear yet for which conditions soil moisture retrieval is actually possible). The Google Earth Engine tool could be improved by adding a masking for certain land covers or vegetation indices.

2. Even for conditions in which Sentinel-1 observations are sensitive to soil moisture, the Sentinel-1 observations are also sensitive to changes of the vegetation cover (for example during the growing season). These effects are not considered in the change detection algorithm.

3. Soil moisture retrievals are not reliable when the Sentinel-1 observations are made under frozen conditions or wet snow conditions. No filtering for these conditions has been applied in the current implementation in Google Earth Engine.

4. The Sentinel-1 observations along a single acquisition orbit are cut in smaller tiles (smaller images). In some cases the images covering the study area (`Area_small`) will be cut in multiple tiles. The adjacent tiles in one acquisition orbit have almost the same date-time of acquisition. This is for example the case for the Raam region for many orbits. By default we do not mosaic multiple tiles that cover `Area_small`, because this takes a long computation time. By defining `MOSAIC = true` (Line 15), multiple images that cover the area of interest on `Date_image` are mosaiced before mapped. For calculation of the graphs images are not mosaiced, so when you export the graph’s data you may see multiple values for one day. The table that can be exported also lists the number of pixels, which could be used for calculation of a weighted average for the total area of interest.

5. For this version of the Google Earth Engine implementation, we used BOFEK version 2.1. Compared to earlier versions of BOFEK one value of the Mualem-Van Genuchten parameters has been changed. We contacted Wageningen Environmental Research to find out which parameter value is correct, but we do not have a definitive answer yet. This parameter affects

the soil moisture content that is obtained for the wilting point for certain areas in the Netherlands (part of the clay regions).

Websites

https://developers.google.com/earth-engine/sentinel1 (last accessed on 26 February 2019)

https://developers.google.com/earth-engine/ (last accessed on 26 February 2019)

www.owas1s.nl (last accessed on 26 February 2019)

https://sentinel.esa.int/web/sentinel/user-guides (last accessed on 26 February 2019)

https://www.wur.nl/nl/show/Bodemfysische-Eenhedenkaart-BOFEK2012.htm (BOFEK2012 map, downloaded on 26 February 2019)

References

Bauer-Marschallinger, B., Freeman, V., Cao, S., Paulik, C., Schaufler, S., Stachl, T., Modanesi, S., Massari, C., Ciabatta, L., Brocca, L., Wagner, W., 2019. Toward Global Soil Moisture Monitoring With Sentinel-1: Harnessing Assets and Overcoming Obstacles. IEEE Trans. Geosci. Remote Sens. 57, 520–539. https://doi.org/10.1109/TGRS.2018.2858004

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sens. Environ. 202, 18–27. https://doi.org/10.1016/j.rse.2017.06.031

Lievens, H., Verhoest, N.E.C., De Keyser, E., Vernieuwe, H., Matgen, P., Álvarez-Mozos, J., De Baets, B., 2011. Effective roughness modelling as a tool for soil moisture retrieval from C- and L-band SAR. Hydrol. Earth Syst. Sci. 15, 151–162. https://doi.org/10.5194/hess-15-151-2011

Mladenova, I.E., Jackson, T.J., Bindlish, R., Hensley, S., 2013. Incidence Angle Normalization of Radar Backscatter Data. IEEE Trans. Geosci. Remote Sens. 51, 1791–1804. https://doi.org/10.1109/TGRS.2012.2205264

Torres, R., Snoeij, P., Geudtner, D., Bibby, D., Davidson, M., Attema, E., Potin, P., Rommen, B., Floury, N., Brown, M., Traver, I.N., Deghaye, P., Duesmann, B., Rosich, B., Miranda, N., Bruno, C., L'Abbate, M., Croci, R., Pietropaolo, A., Huchler, M., Rostan, F., 2012. GMES Sentinel-1 mission. Remote Sens. Environ. 120, 9–24. https://doi.org/10.1016/j.rse.2011.05.028

Ulaby, F.T., Moore, R.K., Fung, A.K., 1982. Microwave Remote Sensing: Active and Passive. Volume II - Radar Remote Sensing and Surface Scattering and Emission Theory. Addison-Wesley Publishing Company, Reading, MA USA.

Van der Velde, R., Benninga, H.F., Pezij, M., Augustijn, D.C.M., Carranza, C.D.U., Van der Ploeg, M.J., Hendriks, D., 2018. Bodemvocht uit satellietdata: wat kan de Nederlandse waterbeheerder ermee? H2O 1–11.

Van der Velde, R., Salama, M.S., Eweys, O.A., Wen, J., Wang, Q., 2015. Soil Moisture Mapping Using Combined Active/Passive Microwave Observations Over the East of the Netherlands. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 8, 4355–4372. https://doi.org/10.1109/JSTARS.2014.2353692

Van der Velde, R., Su, Z., 2009. Dynamics in land-surface conditions on the Tibetan Plateau observed by Advanced Synthetic Aperture Radar (ASAR). Hydrol. Sci. J. 54, 1079–1093. https://doi.org/10.1623/hysj.54.6.1079

Van Genuchten, M.T., 1980. A Closed-form Equation for Predicting the Hydraulic Conductivity of

Unsaturated Soils. Soil Sci. Soc. Am. J. 44, 892–898. https://doi.org/10.2136/sssaj1980.03615995004400050002x

Wagner, W., Lemoine, G., Borgeaud, M., Rott, H., 1999. A Study of Vegetation Cover Effects on ERS Scatterometer Data. IEEE Trans. Geosci. Remote Sens. 37, 938–948. https://doi.org/10.1109/36.752212

Wösten, J.H.M., De Vries, F., Hoogland, T., Massop, H., Veldhuizen, A., Vroon, H., Wesseling, J., Heijkers, J., Bolman, A., 2013. BOFEK2012, de nieuwe, bodemfysische schematisatie van Nederland (in Dutch) [WWW Document]. Alterra Rep. 2387. URL http://edepot.wur.nl/247678 (accessed 9.27.18).