



**HAL**  
open science

## Automated Processing of Sentinel-2 Products for Time-Series Analysis in Grassland Monitoring

Tom Hardy, Marston Domingues Franceschini, Lammert Kooistra, Marcello  
Novani, Sebastiaan Richter

► **To cite this version:**

Tom Hardy, Marston Domingues Franceschini, Lammert Kooistra, Marcello Novani, Sebastiaan Richter. Automated Processing of Sentinel-2 Products for Time-Series Analysis in Grassland Monitoring. 13th International Symposium on Environmental Software Systems (ISESS), Feb 2020, Wageningen, Netherlands. pp.48-56, 10.1007/978-3-030-39815-6\_5 . hal-03361882

**HAL Id: hal-03361882**

**<https://inria.hal.science/hal-03361882>**

Submitted on 1 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Automated processing of Sentinel-2 products for time-series analysis in grassland monitoring

Tom Hardy<sup>1</sup>, Marston Domingues Franceschini<sup>1</sup>, Lammert Kooistra<sup>1</sup>,  
Marcello Novani<sup>1</sup>, Sebastiaan Richter<sup>2</sup>

<sup>1</sup> Laboratory of Geo-Information Science and Remote Sensing, Wageningen University, P.O. Box 47, 6700 AA Wageningen, The Netherlands; Emails: tom.hardy@wur.nl, marston.franceschini@wur.nl, lammert.kooistra@wur.nl, marcello.novani@wur.nl

<sup>2</sup> Versuchs- und Bildungszentrum Landwirtschaft Haus Riswick, Elsenpass 5, 47533 Kleve, Germany; Email: sebastiaan.richter@lwk.nrw.de

**Abstract.** Effective grassland management practices require a good understanding of soil and vegetation properties, that can be quantified by farmers' knowledge and remote sensing techniques. Many systems have been proposed in the past for grassland monitoring, but open-source alternatives are increasingly being preferred. In this paper, a system is proposed to process data in an open-source and automated way. This system made use of Sentinel-2 data to support grassland management at Haus Riswick in the region around Kleve, Germany, retrieved with help of a platform called Sentinelsat that was developed by ESA. Consecutive processing steps consisted of atmospheric correction, cloud masking, clipping the raster data, and calculation of vegetation indices. First results from 2018 resembled the mowing regime of the area with four growing cycles, although outliers were detected due to a lack of data caused by cloud cover. Moreover, that year's extremely dry summer was visible in the time-series pattern as well. The proposed script is a primary version of a processing chain, which is suitable to be further expanded for more advanced data pre-processing and data analysis in the future.

**Keywords:** Grassland monitoring, Open-source system, Sentinel-2, Cloud cover, Time-series analysis.

## 1 Introduction

Efficient grassland management is beneficial to the cultivation of crops in a sustainable and cost-effective way. It is therefore important to quantify the spatial and temporal variation of such lands, in order to get a better understanding of soil and vegetation properties such as nitrogen (N) uptake and the health status of grasslands [1]. One way to quantify such features is based on farmers' experience, but this is increasingly being complemented by technologies such as remote sensing, which makes use of data derived from satellite and airborne based platforms [2].

In recent years, many systems have been proposed and developed for the purpose of (grass)land monitoring, such as the *groenmonitor* (green monitor) initiated by Wageningen Environmental Research, aimed at providing the most recent vegetation information that was extracted from satellite imagery [3]. Another platform is Descartes Labs, which collects and processes open satellite data on a daily basis to prepare it for further analysis such as time-series to determine land cover change over time [4]. However, these initiatives are on a commercial basis for which it is hard to gain insight in the underlying processing chains, making open-source based platforms more and more attractive. One such platform is Google Earth engine aimed at environmental data-analysis on a planetary scale, making use of open-source data from satellite platforms such as Landsat and Sentinel [5]. Although it is a very promising platform to perform big data analysis in remote sensing, it is required to have programming skills to some extent, for instance in JavaScript, in order to perform such analyses.

This paper describes an alternative for developing a grassland monitoring system in an automated way, aimed at providing stakeholders with up-to-date information about the condition and cultivation status of grasslands throughout a growing season in an open-source platform. Stakeholders include for instance farmers, plant breeders, or companies providing agricultural services to those farmers and breeders. Moreover, this application could also be relevant for researchers at universities or research companies, to be able to use the obtained data and information for further research in relation to crop monitoring or environmental protection. The proposed information consists of time-series graphs and output maps of vegetation indices throughout a growing season for any chosen grassland (or land with other agricultural purposes) worldwide. For this paper, this information is derived from images obtained by Sentinel-2 satellites, which are part of an online and open source platform provided by the European Space Agency (ESA).

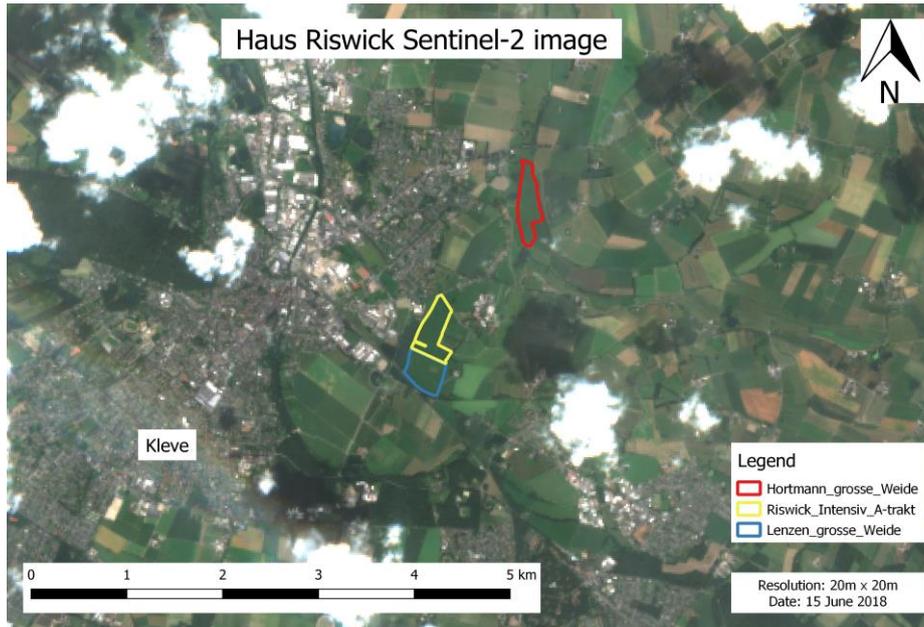
## 2 Methodology

For the study area described in section 2.1, the proposed processing chain started with querying and harvesting Sentinel-2 satellite data (explained in sections 2.2 and 2.3). Next processing steps consisted of an atmospheric correction and extraction of these data for each individual parcel, which is further explained in sections 2.4 and 2.5. The final part of the workflow involved calculation and visualization of vegetation indices and cloud cover percentages for each given date (described in section 2.6).

### 2.1 Study area

The study area for this case study was Haus Riswick, located just east of the city of Kleve, Germany (Fig. 1). Haus Riswick is a teaching and research institution facilitated by the *Landwirtschaftskammer* (chamber of agriculture) from Nordrhein-Westfalen. The soil in that area consists of sandy, clay loam [6]. Data provided by Haus Riswick were geoJSON files representing the field boundaries from (experimental) grassland parcels located at Haus Riswick, as well as harvest data obtained

during the days of mowing including its grassland yield expressed in dry matter content per hectare. A geoJSON is a data exchange format based on JavaScript Object Notation (JSON), which is able to contain a variety of geospatial attributes [7]. For this case study, the data from the field *Lenzen Große Weide* (the blue polygon in Fig. 1) were used.



**Fig. 1.** The region where Haus Riswick is located (east of the city of Kleve), including three of its grassland parcels. Only data from the field *Lenzen Große Weide* were used in this paper.

## 2.2 Sentinel-2 satellite data

Besides the field data from Haus Riswick, earth observation data recorded with Sentinel satellites were addressed for this application. Sentinel satellites are being developed especially for the Copernicus program, one of ESA's earth observation programs aimed at improving environmental management and monitoring the impact of climate change worldwide [8]. For this program, a number of satellites are launched, such as weather satellites and satellites for land- and ocean monitoring. This project made use of data obtained from the two Sentinel-2 satellites (2A and 2B, which both sense the earth in parallel), designed to obtain high resolution (currently 60m, 20m and 10m) optical images for land monitoring worldwide [9].

One way to obtain Sentinel-2 imagery would be by means of the Open Access Hub, which is part of the ESA Copernicus program. This platform has been developed with the aim to offer free and open access to data recorded by the different Sentinel satellites. Those data can be viewed and retrieved one by one via an interactive web viewer [10], based on a number of search criteria such as sensing period, sensing

mode, satellite platform, and product type. This may be useful for anyone who is occasionally interested in only one or two images for a specific location. However, when a larger number of images is needed, this process becomes laborious and time-consuming. In that case, a platform such as the Sentinelsat Application Programming Interface (API) [11], which is aimed at being used in a scripting interface to query and harvest Sentinel data, could be a more effective approach.

### 2.3 Harvesting Sentinel-2 data with Sentinelsat API

For the application in this case study, the data were accessed with help of the Sentinelsat API used in Jupyter notebooks, which is an open-source platform for interactive computing, coding and data analysis in programming languages such as R and Python [12]. The API was used to harvest, (pre-)process and analyze all input data in an automated way. Based on parameters and input data such as a given range of dates, the field's geoJSON file providing the geographical extents, cloud cover percentage, and other (meta)data, large batches of Sentinel-2 products could be downloaded at a time. Those products consisted of images with a size of  $100 \times 100$  km<sup>2</sup> for a region in the world matching the location of the input geoJSON file. For most areas in the world, two or three images each week are recorded by either of the two Sentinel-2 satellites. For each of those days, a check was made to see whether a level-2A or a level-1C product was available for the location. A level-1C product contains top-of-atmosphere (TOA) reflectance, while level-2A products contain bottom-of-atmosphere (BOA) reflectance. It was preferred to make use of level-2A imagery, since level-1C products gave inconsistent analysis results because of distortions in atmospheric reflectance. However, if a level-2A image was not available for a given date, a level-1C product was downloaded nonetheless. Consecutively, such product was processed to level-2A with help of the Sen2cor tool.

### 2.4 Atmospheric correction with sen2cor processing tool

The sen2cor tool has been developed as part of the ESA STEP science toolbox and has the purpose to perform pre-processing on level-1C (TOA) images based on a scene classification, and atmospheric, terrain and cirrus correction, leading to Level-2A (BOA) products [13]. The sen2cor tool is a standalone package and can be freely obtained from the ESA STEP website [14]. The package contains a batch file that was accessed through a python script to automatically process level-1C products to level-2A images.

### 2.5 Extracting data based on field boundaries

After downloading and levelling up of Sentinel-2 products, all raster bands for each date with a resolution of 20m were clipped according to the *Lenzen Große Weide* field's geoJSON file for from Haus Riswick. For a resolution of 20m, the RGB bands, three Red-edge bands, one NIR band, two SWIR bands and a cloud mask were available. The function GDALWarp, part of the GDAL library for fast processing of raster

and vector data [15] was used to clip all raster bands according to the input field boundary. Besides raster bands and the geoJSON representing this boundary, other input parameters for GDALWarp were for instance input and output coordinate reference system, output format (such as a GeoTIFF file), output resolution and resampling method of the pixels.

## 2.6 Calculation and visualization of vegetation indices

From the different Sentinel-2 bands, vegetation indices were calculated. For the initial version of the processing chain, three indices were included: NDVI, WdVI, and a Red Edge index that made use of the sharp increase in reflectance between Red and NIR, useful to quantify leaf chlorophyll content in crops [16]. Its equation is as follows, where  $RE_{band7}$  and  $RE_{band5}$  represent Sentinel-2 bands with wavelengths in respectively the higher and lower regions of the Red Edge slope:

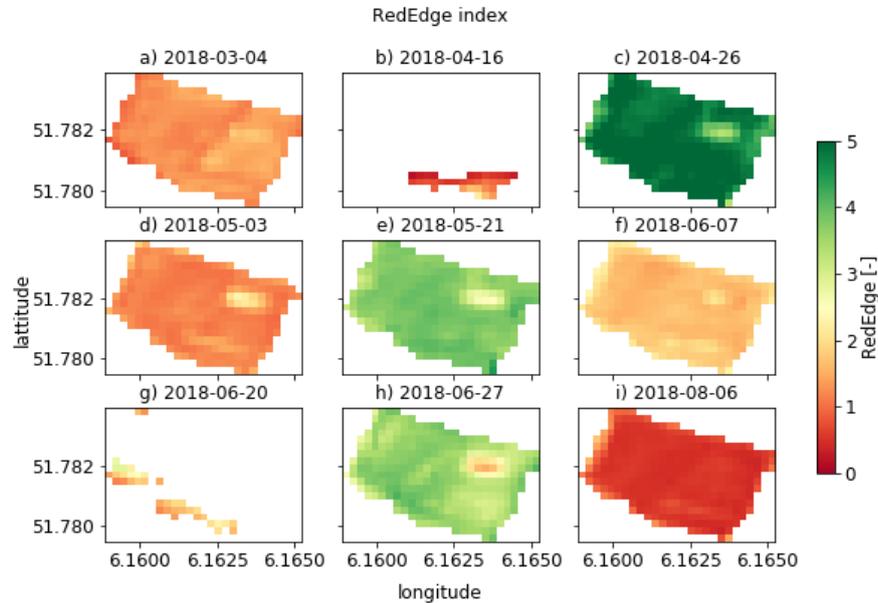
$$CI_{red\ edge} = \frac{RE_{band7}}{RE_{band5}} - 1 \quad (1)$$

For each Sentinel-2 product, an extra masking band was available. This band was a classification raster with classes representing cloud probability, cloud shadow, vegetation, unvegetated pixels, water, cirrus, and snow. The vegetation index raster was masked according to this masking band [17]. This information was visualized by raster maps for each date in range of dates on which either of the two Sentinel-2 satellites recorded data (given that the maximum cloud cover was not exceeded). For presentation purposes, these maps were compiled in GIF animations showing the crop growth over time in a more dynamic way.

For each date in the input date range, the vegetation index mean and standard deviation were calculated based on the Sentinel-2 pixels located within the field boundary. The cloud cover percentage was calculated by dividing the number of cloud-masked pixels by the total number of pixels within the field boundary. Raster datasets containing more cloud-masked pixels than the selected cloud cover threshold were discarded. For the selected range of dates, vegetation index means and standard deviations, as well as cloud cover were visualized by means of time-series charts.

## 3 Results

The aim of the proposed application is to generate relevant information from open-source data, based on a limited number of input parameters. Ideally, those parameters should include start- and end date, which vegetation index to calculate as described in section 2.4, and maximum cloud cover percentage for visualization of results. However, in order to guarantee a smoother data processing workflow, additional parameters such as main working directory, folder paths to input and output locations, and the location of the field boundary shape- or geoJSON file should be provided in the Python script as well.

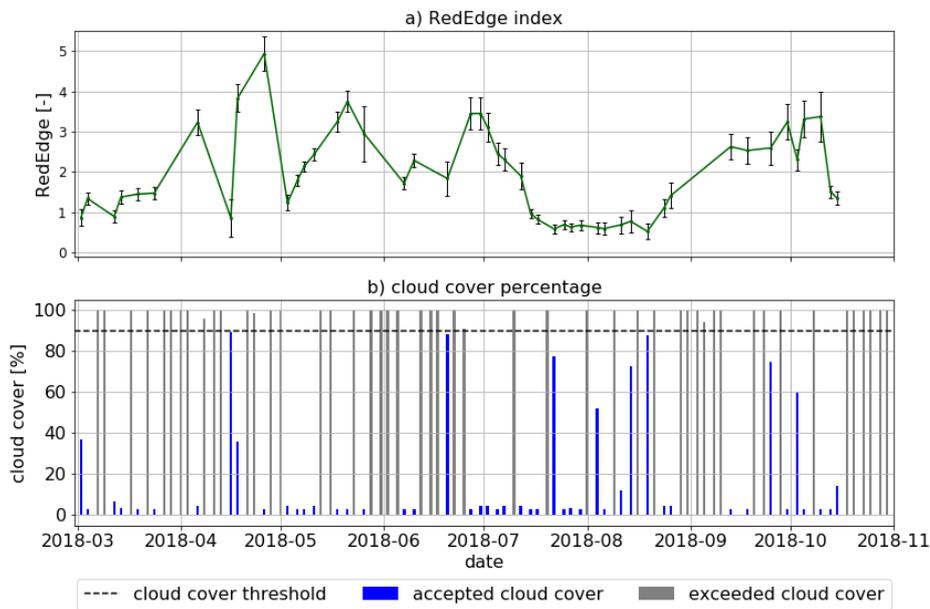


**Fig. 2.** Red Edge on March 4 (a), April 16 (b), April 26 (c), May 3 (d), May 21 (e), June 7 (f), June 20 (g), June 27 (h), August 6 (i), 2018

Based on these input parameters and the workflow in the script, information as depicted in Fig. 2 and Fig. 3 was created. Fig. 2 shows the Red Edge index calculated according to eq. 1, for nine selected days during the growing season of 2018. For this specific case study, that index ranged roughly between values of 0 and 5, where red colors represent values closer to zero and green colors imply values closer to 5. For most maps depicted in Fig. 2, 100 percent (so zero percent cloud cover) of pixels were available, but Fig. 2b and Fig. 2g were greatly hindered by cloud cover. In addition, almost in the north-east of this field a divergent patch of pixels is visible in some of the maps, which is a location with experimental plots having a different mowing regime than the rest of the field. The mowing dates for the field *Lenzen Große Weide* in 2018 were April 26, May 26, July 12, and October 9, indicating a season with four growing cycles. On the one hand, the data for the map in for instance Fig. 2c was recorded just before harvesting of the field, clearly depicted by high values in Red Edge index. On the other hand, the data for Fig. 2d was taken not long after mowing, which is represented by lower Red Edge values. Similar patterns were also observed for the dates close to the other three moments of harvesting. Lastly, the map in Fig. 2i shows very low values in Red Edge, which was caused by the extremely dry summer of 2018.

Fig. 3a shows time-series containing the mean and standard deviation of the selected vegetation index, and Fig. 3b depicts the cloud cover percentage for the given range of dates. The mowing dates described earlier are also resembled in Fig. 3a, since large drops in Red Edge are visible for those moments in the growing season. However, also large decreases in Red Edge are visible on for instance April 16, as

well as a pattern containing very low values for the months July and August. One explanation for the first issue is the large influence of cloud cover, while the second issue was caused by the very dry summer of 2018. Fig. 3b shows the cloud cover range for each date in the growing season, with grey values exceeding the maximum cloud cover percentage and blue values falling below this threshold. For this case study, the cloud cover threshold was set to 90 percent. Only data recorded on days whose cloud cover did not exceed this threshold were used to produce a time-series output as shown in Fig. 3a.



**Fig. 3.** Red Edge index (a) and cloud cover percentage (b) from March 1 till October 31, 2018

## 4 Discussion

In this paper, an alternative was proposed with a first version of a processing chain for extracting, processing and analyzing satellite-based data for grassland monitoring in an automated and open-source way, aimed at stakeholders resembling agribusiness companies and researchers to support their services and research practices. The development and primary testing of this processing chain was mainly conducted on a university's notebook. However, a notebook's or PC's processing memory and storage capacity are limited, which is one reason to upscale the processing workflow to a cloud computing environment for more efficient data processing and analysis in the future. Moreover, this system could still be enhanced to perform more advanced data (pre)processing and analysis, for instance by creating a more elaborated atmospheric correction algorithm and the inclusion of additional vegetation indices. For instance, atmospheric correction was merely performed with help of the `sen2cor` tool developed

by ESA. A great variability in radiometry was observed comparing Sentinel images of multiple days. Therefore, many processing parameters could have been changed in configuration files (xml files as part of a Sentinel product's metadata) to be used in the `sen2cor` tool for each day separately. These options were considered, but were not included in the pre-processing chain, since this was tough to implement in an automated way. An alternative to the `sen2cor` tool could have been MACCS/MAJA, which has a more elaborated workflow in comparison to the `sen2cor` tool, since it includes more advanced weather analysis, cloud detection and atmospheric correction in its algorithm [18]. In addition, ESA has been started to process all Sentinel L1C data to L2A in retrospect, so the `sen2cor` tool might become obsolete in the future.

The next step was to discard vegetation images with a cloud cover more than a given threshold (90% was used for this case study, see Fig. 3b). One way to deal with this issue would be to lower the cloud cover threshold, which would lead to more accurate vegetation index output for the remaining data, but also to a loss of data not meeting the cloud cover threshold requirement. For instance, the maps in Fig. 2b and Fig. 2d were greatly influenced by cloud cover, which is resembled by the outliers in Fig. 3a for the dates on April 16 and June 20. In conclusion, the cloud cover percentage threshold is a trade-off that the end-user of this monitoring system should decide upon. Another way to improve this process would be to take a sequence of images in time as input, and systematically replace disturbed pixels from one image by clean pixels from a subsequent or previous image. Clean pixels are pixels that do not containing cloud cover or NA values. This replacing process could for example be performed by the `sen2three` tool, which was developed as part of the ESA STEP program as well [19]. However, this tool takes a sequence of Sentinel L2A images and merges them into one clean composite image. Since data was limited already, this could have had a disadvantageous effect on time series output such as in Fig. 3a.

## 5 Conclusions

The main goal of this paper was to develop and provide insight in the processing workflow of a grassland monitoring system in an open-source and automated way. In such workflow, processing steps are required to get from input data to output representing useful information for grassland management practices, and to compare that output to information derived from airborne or ground based data.

The higher the cloud cover threshold, the more data is available to create a time-series, but also the lower the accuracy becomes. Lowering this threshold leads to a loss of data but also to an increase in accuracy. This is a trade-off that should be decided upon by the end-user of the monitoring system.

Despite the outliers caused by a lack of pixels due to cloud cover, first results gave insights about the mowing regime over time, which could support optimizing and managing a grassland system in a more effective way. These results and data could be useful as well to elaborate on in further research.

The proposed script was a preliminary version of a processing change, which is ready to be further enhanced and expanded with coding for more advanced data pre-processing and data analysis in the future.

## 6 Further research

- Further develop processing chain for more elaborate pre-processing and analysis steps such as atmospheric correction and the inclusion of extra vegetation indices.
- Further develop processing chain to make it suitable to include airborne based data.
- Translate time-series information into crop biomass information for the different cycles in a growing season.
- Upscale the processing chain to a cloud computing environment for more efficient data processing and analysis.

## Acknowledgements

This work was supported by the SPECTORS project (143081), which is funded by the European cooperation program INTERREG Deutschland-Nederland.

## References

1. Capolupo A, Kooistra L, Berendonk C, et al (2015) Estimating plant traits of grasslands from UAV-acquired hyperspectral images: A comparison of statistical approaches. *ISPRS Int J Geo-Information*. <https://doi.org/10.3390/ijgi4042792>
2. Mulla DJ (2012) Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosyst Eng* 114:358–371. <https://doi.org/http://dx.doi.org/10.1016/j.biosystemseng.2012.08.009>
3. Wageningen Environmental Research Groenmonitor. <http://www.groenmonitor.nl/>
4. DescartesLabs DescartesLabs. <https://www.descarteslabs.com/>
5. Gorelick N, Hancher M, Dixon M, et al (2017) Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sens Environ* 202:18–27. <https://doi.org/https://doi.org/10.1016/j.rse.2017.06.031>
6. Haus Riswick Haus Riswick. <http://www.riswick.de>
7. Butler H, Daly M, Doyle A, et al (2016) The GeoJSON Format. RFC Editor
8. ESA ESA Copernicus. [https://m.esa.int/Our\\_Activities/Observing\\_the\\_Earth/Copernicus/Overview3](https://m.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview3)
9. ESA ESA Copernicus data products. [https://www.esa.int/Our\\_Activities/Observing\\_the\\_Earth/Copernicus/Sentinel-2/Data\\_product](https://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-2/Data_product)
10. ESA Copernicus Scihub. <https://scihub.copernicus.eu/dhus/#/home>
11. ESA Copernicus Scihub API Hub.

- <https://scihub.copernicus.eu/twiki/do/view/SciHubWebPortal/APIHubDescription>
12. Wille M, Clauss K Sentinelsat. <https://sentinelsat.readthedocs.io/en/stable/#>
  13. ESA S2A processing user guide. <https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels/level-2>
  14. ESA ESA STEP. <http://step.esa.int/main/third-party-plugins-2/sen2cor/>
  15. Warmerdam F, Rouault E GDALWarp. <https://gdal.org/programs/gdalwarp.html>
  16. Gitelson AA, Viña A, Ciganda VS, et al (2005) Remote estimation of canopy chlorophyll content in crops
  17. ESA S2A processing algorithm. <https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm>
  18. CESBIO multitemp MACCS/MAJA tool. <https://labo.obs-mip.fr/multitemp/maccs-how-it-works/>
  19. ESA Sen2Three tool. <http://step.esa.int/thirdparties/sen2three/1.1.0/sen2three-1.1.0.html/doc/index.html>