"sen2r": An R toolbox for automatically downloading and preprocessing Sentinel-2 satellite data

Article *in* Computers & Geosciences · March 2020

DOI: 10.1016/j.cageo.2020.104473

"sen2r": an R toolbox for automatically downloading and preprocessing Sentinel-2 satellite data [1]

Luigi Ranghetti[a,*], Mirco Boschetti[a], Francesco Nutini[a], Lorenzo Busetto[a]

[a] *Institute for Remote Sensing of Environment, Consiglio Nazionale delle Ricerche, via Bassini 15, I-20133 Milano, Italy*

Abstract

sen2r is a scalable and flexible R package to enable downloading and preprocessing of Sentinel-2 satellite imagery via an accessible and easy to install interface. It allows the execution of several preprocessing steps which are commonly performed by Sentinel-2 users: searching the Sentinel-2 archive for datasets available over a spatial area of interest and in a defined time window, downloading them, applying the Sen2Cor atmospheric correction algorithm to compute surface reflectances, merging adjacent tiles, performing geometric transformations, applying a cloud mask, computing spectral indices and colour images. The package is designed to be accessible to a range of users, from beginners to skilled R users. It comes with a Graphical User Interface, which can be used to set the processing parameters and launch processing operations: this feature makes sen2r accessible also for novices with limited programming experience. High-level R functions, which enable customised image processing workflows and control over intermediate steps, can be useful to experienced remote sensing researchers. Thanks to those functions it is possible to easily schedule automatic processing chains, so to manage massive processing operations. This paper describes the main characteristics, functionalities and performance of the package and highlights its usefulness as the operational back-end of service-oriented architectures, as illustrated by the Saturno project.

Keywords: Data Processing, Software Engineering, Parallel And High-Performance Computing, Remote Sensing

1. Introduction

The usefulness of satellite images for monitoring and investigating the characteristics of the Earth's surface depends on their spatial and temporal resolution. This is particularly true for applications aimed at monitoring rapidly-changing systems, such as agronomic/phenological studies (Zhang et al., 2009; Isaacson et al., 2012) or operational services for detection of sudden changes in land cover status (Hansen et al., 2008; Ju and Roy, 2008). However, spatial resolution and temporal frequency are often complementary properties of satellite data: coarse resolution images, like those acquired by the Moderate Resolution Imaging Spectroradiometer (MODIS) sensors, are usually characterised by a short revisiting time, particularly suitable for time series analysis but limited in their usefulness for the discrimination of specific targets (Bruce et al., 2006; Shi et al., 2018). Conversely, multispectral medium resolution datasets, like those historically acquired with the Landsat TM, ETM+ and OLI sensors, allow much better target discrimination but are characterised by a revisiting time often too long for detecting rapid land cover changes or reconstruct phenological trends (Reed et al., 2009).

In this framework, a turning point was marked by the availability of data acquired by the Sentinel-2 (S2) multispectral sensors, recently launched as part of the European Copernicus programme. Sentinel-2 features a decametric resolution (10 m, 20 m or 60 m, depending on the spectral band) and a large swath of 290 km. Each of the S2 sensors grants a revisiting time of 10 days at the equator, which reduces to 5 days considering the combination of both the Sentinel-2A and Sentinel-2B satellites (launched respectively 3 June 2015 and 7 March 2017 – European Space Agency 2015), or even shorter on areas covered by multiple S2 orbits. These features are providing new opportunities in remote sensing applications requiring both high temporal and spatial resolution (Drusch et al., 2012). Examples of this include analysis of crop phenology at intra-field scale (Campos-Taberner et al., 2016, 2017; Busetto et al., 2017; Boschetti et al., 2018 among others), study of vegetation dynamics in heterogeneous conditions (Veloso et al., 2017; Vrieling et al., 2018; Stendardi et al., 2019), fine resolution land cover detection (Immitzer et al., 2016; Lefebvre et al., 2016; Paul et al., 2016; Pesaresi et al., 2016; Radoux et al., 2016; Belgiu and Csillik, 2018), fire monitoring and mapping (Huang et al., 2016; Verhegghen et al., 2016).

* Corresponding author
Email address: ranghetti.l@irea.cnr.it (Lorenzo Busetto)

Postprint accepted by Computers and Geosciences (DOI: 10.1016/j.cageo.2020.104473). *Accepted 8th March 2020*

1.1. *Current possibilities for accessing and analysing Sentinel-2 data*

The European Space Agency (ESA), in the framework of the Copernicus European Programme, manages an online data hub (`https://scihub.copernicus.eu/dhus`) to interactively search for available Sentinel images and download them; access to data is free, upon registration. Data can also be retrieved via scripting using the SciHub's Advanced Programming Interface (API) (`https://scihub.copernicus.eu/userguide/8BatchScripting`): this has promoted the development of many different services and tools which allow searching and downloading of S2 data; a very exhaustive list of them is provided by Clauss (2019).

S2 data are provided by ESA as ZIP-compressed archives in SAFE format (Standard Archive Format for Europe). Each archive contains surface reflectance data stored as JPEG2000 multispectral rasters, corresponding to specific 110×110 km tiles of the Sentinel-2 tiling grid (see `https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products` for details). Users can download either L1C data, containing Top of Atmosphere (TOA) reflectances for 12 bands encompassing the 440 to 2200 nm spectral range, or L2A data, containing Bottom of Atmosphere (BOA) reflectances for 11 bands – band 10 is absent in L2A products – derived by applying the Sen2Cor algorithm to TOA data. Additionally, L2A datasets contain a Scene Classification Map (SCL) raster layer computed by Sen2Cor as a side-product. The SCL subdivides the image in 12 classes: "no data", "saturated or defective", "dark area pixels", "cloud shadows", "vegetation", "not vegetated", "water", "unclassified and cloud low probability", "cloud medium probability", "cloud high probability", "thin cirrus" and "snow".

Although as previously stated many solutions exist for searching and downloading S2 data, the number of tools available for their processing is more limited. Some of them are dedicated to atmospheric correction and cloud detection, like Sen2Cor (Richter et al., 2012; Main-Knorn et al., 2017), MAJA (Hagolle et al., 2010; Baetens et al., 2019), ARCSI (`https://www.arcsi.remotesensing.info`) and s2cloudness (`https://github.com/sentinel-hub/sentinel2-cloud-detector`).

Among tools allowing users to perform a wider range of processing operations, the most complete is the Sentinel-2 Toolbox provided by ESA (European Space Agency, 2018b), written in Java and released under the GNU GPL license. It includes the aforementioned Sen2Cor processor for the production of level 2A bottom-of-atmosphere reflectances, the Sen2Three processor for the generation of level-3 products (composites of cloud-free images over a defined time period) and the processor of level-2B biophysical variables (like LAI and fAPAR); moreover, it allows computing a moderate number of Spectral Indices (SIs) starting from reflectance data. It can be used from the command line or from the SNAP (Sentinel Application Platform) interface, which includes the Sentinel-2 Toolbox applications within a more user-friendly desktop interface. Nevertheless, the batch processing of a high number of images and the automatic scheduling of processing tasks can not be easily carried out.

Another relevant tool is Sen2-Agri (European Space Agency, 2018a), an open-source system which can be implemented on a user-deployed server to process S2 data. It is mainly written in Bash (processing tools), HTML and JavaScript (interface). It allows generating a set of products mainly useful for agronomic applications: monthly composite level-3A images, NDVI and LAI maps (level-3B), monthly cropland masks (level-4A) and crop type maps (level-4B). It includes a HTML user interface, and can be used to schedule automatic processing chains over large areas. The main cons limiting its usefulness are in our opinion related to the lack of customisability of the products (i.e. SIs others than NDVI) and to the hardware requirements (80 GB for system installation, 10 TB for image processing, 64 GB RAM), which make this system suitable for a server implementation but not for desktop use.

Among third-party cloud-based services, Google Earth Engine (Gorelick et al., 2017) deserves to be mentioned. It allows applying standard or customised processing algorithms (written in Javascript or Python) to S2 data. The main advantage of this platform is its very high computation power, achieved thanks to i) the availability of a S2 data catalogue directly accessible from processing machines, and ii) the high availability of CPUs in Google data centres and the efficiency in distributing complex computations. The main disadvantages of this infrastructure are the difficulty of its use for people with limited programming skills and the limitations for accessing outputs generated by custom scripts (data can be transferred to a Google Cloud Storage – paid – or to a Google Drive user space – whose size is generally limited or again subject to fees).

1.2. *A new solution: the sen2r package*

In this manuscript we present the sen2r package, which was created with the objective of providing a suite of processing tools for Sentinel-2 data sufficiently user-friendly to allow its use by end-users with limited programming skills, but at the same time powerful enough to allow more skilled users to automate the processing of S2 data over large spatial/temporal extents.

The package was developed in the R software (R Core Team, 2018). There were several reasons for this choice. First, R is one of the most used open-source languages for data analysis, and its capabilities for spatial data analysis is continuously and rapidly evolving (Lovelace et al., 2019; Bivand, 2019), so many people interested in S2 data preprocessing already uses R and are familiar with its syntax. Moreover, many procedures for high-level spatial processing can already be exploited from R, i.e. using R packages like raster (Hijmans, 2019), sf (Pebesma, 2018) and stars (Pebesma, 2019), or calling external processing programmes like GDAL (GDAL/OGR contributors, 2018) with system calls. Additionally, despite R being natively single-threaded, it easily allows for building routines exploiting parallelised computation through the package parallel (R Core Team, 2018), thus allowing users to exploit the power of modern multi-core infrastructures. Finally, packages for raster processing like raster or stars allow processing large rasters by chunks of lines, without the need to load the whole data matrix in memory. The latter two features are fundamental to implement functions which can efficiently deal with high-frequency time series of large raster files both on "standard" desktop PCs or more powerful server infrastructures. Another advantage offered by R is the possibility to easily create and deploy powerful interactive Graphical User Interfaces based on the Shiny package (Chang et al., 2018).

The main functionalities of sen2r are described in detail in section 2. Basically, the package allows users to easily configure and manage processing chains for creating time series of different raster products derivable from S2 data (surface reflectances, scene classification maps, RGB and SIs images) over a user-specified Area Of Interest (AOI) and time window, and exporting them to an easier to use file format. This can be done using a single R function, `sen2r()`, which can be initialised either by using the aforementioned GUI, or by specifying the required arguments from the command line; intermediate functions are also provided to allow independent execution of specific steps (sections 3.2 and 3.3). Processing a high number of images over a wide area is possible thanks to the scalability of the package: section 2.8 describes the way sen2r can manage complex processing chains.

The outputs of sen2r consist of single-date raster files covering the selected AOI and following a strict naming convention (see section 3.4.1). Multitemporal archives of sen2r outputs can be easily progressively updated as soon as new S2 data are made available (see section 3.3.3) or integrated at a later stage with new products (i.e. an additional SI). These characteristics make sen2r useful for both small / sporadic processing operations as well as for operational back-end of web applications or geoportals, as highlighted by the case study described in section 4.

2. The sen2r processing chain

This section briefly describes the main steps of a typical sen2r processing chain (figure 1). Details about the commands to be used to run it are provided in sections 3.2 and 3.3.

For the sake of simplicity, the main operations performed by sen2r are described in relation to the production of a single set of output products derived from S2 data acquired in a single sensing date. Section 2.8 provides further details on how the package manages multi-date processing chains.

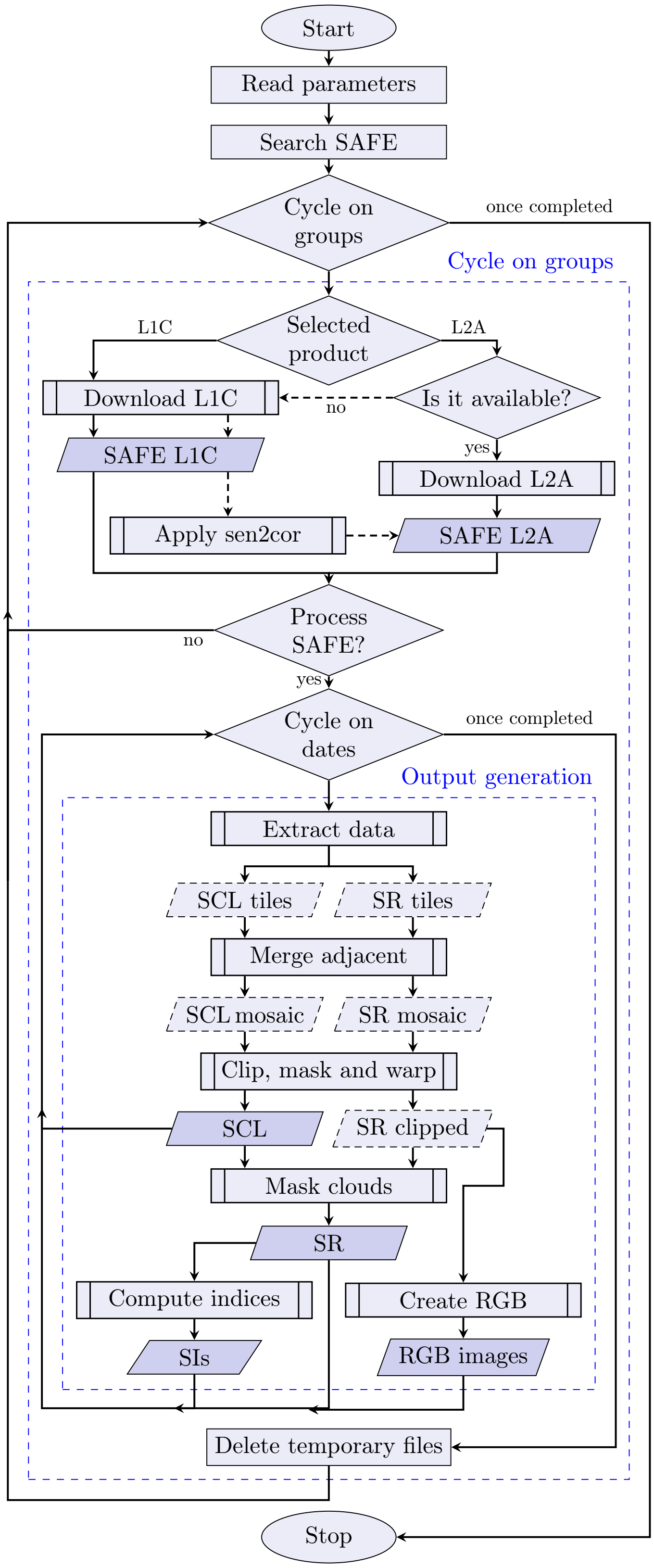


Figure 1: Flow chart of sen2r main processing steps, described in sections 2 (processing steps) and 2.8 (running order). Darker trapezoid refers to output files, while dashed ones to temporary files.

2.1. Read parameters

First of all, sen2r needs to read a set of parameters which describes the desired processing (e.g., spatial-temporal conditions, output products and paths, required geometric transformations and cloud masking). Those parameters can be provided by users in three ways: i) by using the sen2r GUI (see section 3.2); ii) by specifying the path to a JSON text file containing them (which can be exported from the GUI and/or edited by hand); iii) by passing them as arguments to the `sen2r()` function.

2.2. Download data and apply Sen2Cor

The list of Sentinel-2 SAFE archives matching the provided querying parameters is first retrieved; users can choose to search them both online or offline (in the latter case, only SAFE products already existing locally are considered). This list is then used to generate the paths of the expected output files, so that computation of already existing files can be skipped (unless users wish to overwrite them).

The required SAFE archives not found on the user's machine are downloaded from ESA SciHub [2]. Starting from September 2019, part of the Sentinel-2 data (typically, level 2A archives older than 18 months and level 1C older than one year) are stored in the Long Term Archive (LTA; `https://inthub.copernicus.eu/userguide/LongTermArchive`), and no longer immediately available for download. In these cases, sen2r automatically requests the missing data to be restored in the online archive and skips processing dates including archived datasets. The desired processing chain can then be finalised by re-running it after the time required to ESA to put the data back online [3].

By default, if L2A (i.e. atmospherically corrected) images are required for the processing but only L1C are available on the ESA archive, L1C data are downloaded and atmospherically corrected using Sen2Cor to compute the corresponding L2A data. Users can however choose a different behaviour (using only already available L2A data, or always downloading L1C archives and generating the corresponding L2A locally).

2.3. Apply geometric transformations

Rasters of TOA (Top Of Atmosphere) reflectances, BOA (Bottom Of Atmosphere) reflectances and/or SCL layers covering the desired AOI and with the desired output resolution are first of all created. To do this, three intermediate steps are performed:

1. images are extracted from the SAFE archive on the overall extent of the required S2 tiles;
2. images belonging to adjacent tiles of the same frame needed to cover the whole AOI and acquired in the same date are merged;
3. geometric transformations (e.g., reproject, resample, crop and eventually mask pixels outside the output AOI) are performed based on users' settings (see section 3.2.3).

Resulting rasters are stored with a constant user-selected spatial resolution (e.g., if a 10 m output spatial resolution is chosen, S2 bands with lower resolution are automatically resampled to 10 m). Bands 8 ($\sim 690-970$ nm) or 8A ($832-897$ nm) are used conditionally: band 8 if the output resolution is lower than 20 m, band 8A if equal or higher than 20 m.

2.4. Mask clouds

A cloud mask based on the SCL layer (see section 2.3) is applied to the produced BOA/TOA images. Users can choose the classes to be masked [4], as well as the maximum cloud coverage allowed within the defined AOI: if cloud coverage for a given date is higher than this threshold, outputs for that date are not produced.

Moreover, it is possible to perform geoprocessing manipulations on the mask layer (i.e. smoothing and buffering cloud borders): this step allows reducing "salt-and-pepper effects" (i.e. isolated cloudy / non-cloudy pixels within non-cloudy / cloudy areas), and removing low-quality pixels on cloud borders.

2.5. Compute spectral indices

The SIs eventually required by the user are computed from surface reflectances (BOA by default, if available).

2.6. Create RGB images

True- or false-colour 24-bit RGB images based on any desired combination of the original bands can be created from surface reflectances. By default, bands 1 to 5 are stretched in the range $0-0.25$, while bands 6 to 12, in which soil reflectance is usually higher, are stretched in the range $0-0.75$ [5]. Note that, since RGB images are designed as aids for visualisation rather than for data analysis, clouds are not masked.

2.7. Final steps: create thumbnails and remove temporary files

After all the aforementioned steps have been concluded, thumbnails (PNG or JPEG images with a maximum size of 1024×1024 pixels) of the output images can be generated.

Temporary image files needed for processing are finally deleted, so to minimise disk storage usage.

[2] Copernicus data are publicly available, but registration to SciHub is required to download them; therefore users have to provide their credentials in the package, either from the GUI or using the function `write_scihub_login()`.

[3] This functionality is available starting from sen2r v. 1.2.0, which can be currently installed only from GitHub.

[4] The denomination "cloud mask" for this step is therefore not completely correct, since also classes not referring to clouds could be masked, but it is convenient to understand what it is done.

[5] The stretching range can be modified so to obtain brighter or darker outputs, or to emphasise different reflectance ranges.

2.8. Processing order and parallelisation

Due to the high spatial and temporal resolution of S2 data, the execution of a sen2r processing chain may require a long execution time and substantial system resources (both RAM and disk space) when a large AOI and/or a long time period are involved. For this reason, special attention was dedicated to optimisation of system performance, through exploitation of multi-core architectures and careful management of disk usage.

The default processing scheme is aimed at providing a good compromise between processing speed and disk usage. Processing is done as follows: first, the list of required SAFE and output product names is computed (section 2.1); this allows calculating the number of dates which must be processed (n). Then, being m the number of CPU cores available for processing based on system architecture and/or users' choices (see section 3.2.3), the required dates are grouped in g groups, where $g = \lceil \frac{n}{m} \rceil$. Identified groups are then processed one at a time. For each group:

1. all SAFE archives required to process the corresponding dates are downloaded, then Sen2Cor – if required – is applied in parallel using one core per L1C SAFE archive, up to a maximum m cores (section 2.2);

2. once all SAFE archives have been downloaded/processed, the remaining processing operations (sections 2.3 to 2.6) are executed using m parallel R sessions (one core for each date);

3. once all output rasters of the group have been produced, temporary files and SAFE archives are deleted, and processing of the subsequent group is started.

This scheme allows both obtaining a good speed of execution (thanks to multicore processing) and optimising disk usage (since SAFE archives and temporary files required by each group are deleted before new ones are downloaded / produced). RAM requirements can instead be tweaked by adjusting the m number of cores to be used.

Nevertheless, users with specific requirements can select different processing schemes; available options are described below.

- *Process by date:* this allows minimising disk usage (in particular if SAFE archives are deleted after processing). It is similar to the default scheme, but with g forced to n; since each group includes a single date instead than m dates, the disk space occupied by SAFE archives and temporary files is lower. This mode is however generally slower than the default one because parallel computation over dates for products' generation is not possible.

- *Mixed processing:* this allows maximising CPU usage (and thus processing speed). In this case, the cycle on groups is ignored ($g = 1$, i.e. a single group including all the required dates is used), so that all the required SAFE are first of all downloaded and/or produced, and then dates are processed in parallel (i.e., launching a maximum m number of simultaneous R sessions). Although faster than the default mode, this requires all SAFE archives to be downloaded and processed, and all temporary files to be generated, before performing subsequent steps, thus increasing disk space requirements. It is therefore less feasible for processing chains involving long time windows and/or large AOIs.

- *Process step by step:* in this case, both the cycles on groups and dates are ignored. In this case, all SAFEs are first downloaded/processed (eventually exploiting parallelisation of Sen2Cor); then, the processing steps described in sections 2.3 to 2.6 are performed sequentially (i.e. geometric transformations are done for all dates; when this finishes, masks are applied, and so on). Parallelisation over dates is performed in this case within internal functions (e.g., `s2_translate()` or `s2_calcindices()`). This mode is similar to the previous one in terms of disk usage but it is slightly slower, its advantage being the lower RAM requirements.

3. Description of the software

3.1. Installation and dependencies

The stable version of the package is available on CRAN (`https://CRAN.R-project.org/package=sen2r`), while the latest version can be found on GitHub (`https://github.com/ranghetti/sen2r`). A dockerised version of sen2r is also available at `https://hub.docker.com/r/ranghetti/sen2r`.

sen2r can be installed and loaded with the following commands [6]:

```
## Install sen2r
install.packages("sen2r")
library(sen2r)
```

sen2r exploits functionalities provided by several R packages, that are automatically installed alongside the package (see table 1), and by the following runtime executables:

- GDAL (`https://gdal.org`) is an open source library for raster and vector geospatial data formats including a variety of command line utilities for data translation and processing: it is used for some processing operations and to retrieve metadata from SAFE products;

[6] To be sure that all system requirements are satisfied, it is recommended to follow the instructions reported in the online documentation (`http://sen2r.ranghetti.info/articles/installation`).

- Sen2Cor (http://step.esa.int/main/third-party-plugins-2/sen2cor) is a atmospheric correction tool provided by ESA to compute BOA reflectances from Sentinel-2 level 1C products;
- aria2 (https://aria2.github.io) is an optional alternative downloader which can be used to speed-up the download of SAFE archives.

Availability of these runtime dependencies on the local system can be easily checked after installation using a simple GUI [7]:

```
## Check runtime dependencies
check_sen2r_deps()
```

Alternatively, dedicated command line functions (`check_gdal()`, `install_sen2cor()` and `install_aria2()`) can be used for the same purpose.

3.2. *Interactive execution: the sen2r GUI*

`sen2r()` is the main function of the package, allowing setting up and running a complete processing chain.

The easiest way to parameterise the `sen2r()` function is by using the sen2r GUI, a HTML interface built with R Shiny (RStudio, Inc, 2013): this can be done launching this function without specifying any argument (see figure 2):

```
## Run sen2r interactively
sen2r()
```

The GUI uses a standard Shiny Dashboard structure (Chang and Borges Ribeiro, 2018). Its sidebar allows navigating through five main panels (described in sections 3.2.1 to 3.2.5), each allowing setting processing parameters related to a specific aspect. The sidebar can also be used to export the processing parameters to an external JSON file using the "Save options as..." button, allowing users to re-use them in later `sen2r()` executions or within scripts. Previously exported parameters can be restored in the GUI with the "Load options" button. A text file that will store the log of the processing operations can be initialised with the button "Create log...". Finally, the "Launch processing" and "Close without saving" buttons allow closing the GUI, respectively with or without starting the processing chain described in section 2 with current parameters.

3.2.1. *Product selection*

This panel allows setting general processing options concerning type of processing and desired output products. The panel is divided in four boxes.

Type of processing. This box allows users to choose if processing should be limited to downloading SAFE archives (eventually correcting them with Sen2Cor) [8], or if additional preprocessing operations are required.

Products and sensors. This box allows choosing which raster products should be created as outputs, among: surface reflectances (BOA, TOA or both), SCL, SIs and RGB images. Selecting either of the last two options activates specific GUI panels in which options for creation of SIs and RGB can be set. It also allows limiting the processing to a specific satellite (i.e. Sentinel-2A or 2B).

SAFE options. This box allows setting options related to management of SAFE archives. First, paths of the directories where the SAFE archives are stored can be specified. The "Download mode" checkbox allows choosing between connecting to SciHub and downloading all non already available SAFEs (online mode) or using only those already on the users' machine (offline mode). In the first case, some download options (SciHub credentials, downloader to be used, possibility to order SAFEs from LTA and maximum SAFE cloud cover) can be set. It is also possible to specify if already existing SAFE archives should be kept (i.e. not downloaded / reprocessed with Sen2Cor again) or overwritten, and if they should be deleted after processing.

Atmospheric correction options. Starting march 2018, ESA operationally produces both L1C and L2A archives, and is gradually reprocessing older L1C data to create a complete L2A archive; nevertheless, the archive of L2A products is not yet completed, making the possibility to easily produce L2A data from L1C archives still fundamental for the creation of S2 time series. In this box, users can choose if / when Sen2Cor must be run: always (L1C archives are always used to produce L2A), never (only L2A data already on SciHub are used) or only if a L2A image is not available online but the corresponding L1C is. By default, Sen2Cor is launched with default settings, which do not include topographic correction of L1C reflectances: this may be changed by setting "Apply topographic correction?" to "Yes" [9]. A finer customisation of other Sen2Cor parameters can be done using the command line syntax.

3.2.2. *Spatial-temporal selection*

This panel allows specifying the temporal and spatial constraints for the processing. It is divided in two boxes.

[7] On Windows, the function also allows easily installing missing dependencies.

[8] In this first case, most of the selectors for the other processing parameters are disabled.

[9] Doing that and downloading the ESA-CCI data-package ensures that produced L2A archives are consistent with the ones provided by ESA (further details can be found at http://step.esa.int/main/third-party-plugins-2/sen2cor/).

Table 1: List of sen2r R package dependencies.

R package	Reference	Usage
data.table	Dowle and Srinivasan (2019)	Management of large tabular data
digest	Eddelbuettel (2018)	Decimal to hexadecimal conversion
doParallel	Microsoft Corporation and Weston (2018)	Run processes in parallel
foreach	Microsoft Corporation and Weston (2017)	*idem*
geojsonio	Chamberlain and Teucher (2018)	Save spatial extent within the parameter file
httr	Wickham (2018)	Manage http downloads
jsonlite	Ooms (2014)	Import/export parameter files
leaflet	Cheng et al. (2018)	Visualise the AOI and S2 tiles in the GUI
leaflet.extras	Karambelkar and Schloerke (2018)	*idem*
magrittr	Bache and Wickham (2014)	Pipe operator
mapedit	Appelhans and Russell (2018)	Draw a custom AOI in the GUI
raster	Hijmans (2019)	Spatial processing operations on raster files
reticulate	Allaire et al. (2018)	Interface to Python scripts used to search and download S2 SAFE products
sf	Pebesma (2018)	Spatial operations over vector files
shiny	Chang et al. (2018)	Build and manage the GUI
shinydashboard	Chang and Borges Ribeiro (2018)	*idem*
shinyFiles	Pedersen et al. (2018)	*idem*
shinyjs	Attali (2018)	*idem*
shinyWidgets	Perrier et al. (2019)	*idem*
stars	Pebesma (2019)	Collect information on raster files
XML	Lang and the CRAN Team (2019)	Creation/update of the database of spectral indices

Temporal range. This box allows setting the starting and ending dates to be considered in the processing. Setting the selector "Time period type" to "Seasonal" allows downloading a multiannual time series limited to a specified sub-yearly period (e.g., 2016 – 2018 from May to September).

Area of interest. This box allows setting the AOI, by providing the coordinates of a bounding box, selecting a vector file corresponding to the desired extent, or drawing the AOI on a interactive map. After the AOI is defined, it is shown on a map, overlaid with the footprints of S2 tiles intersecting it; this allows users to eventually deselect the tiles they do not want to consider (e.g. because they cover a very small portion of the AOI). Users also need to provide a name for the selected AOI, to be used in the output file names (see section 3.4.1).

3.2.3. Processing options

In this panel, users can set the parameters determining the main characteristics of the output images. The panel is divided in four boxes.

Output files. This box allows setting the main path of the output folder, choosing if files should be put in the same folder or if a subfolder for each product should be created, and deciding the format (GeoTIFF or ENVI) and compression level of the output files. It also allows deciding if thumbnails of the output rasters should be created.

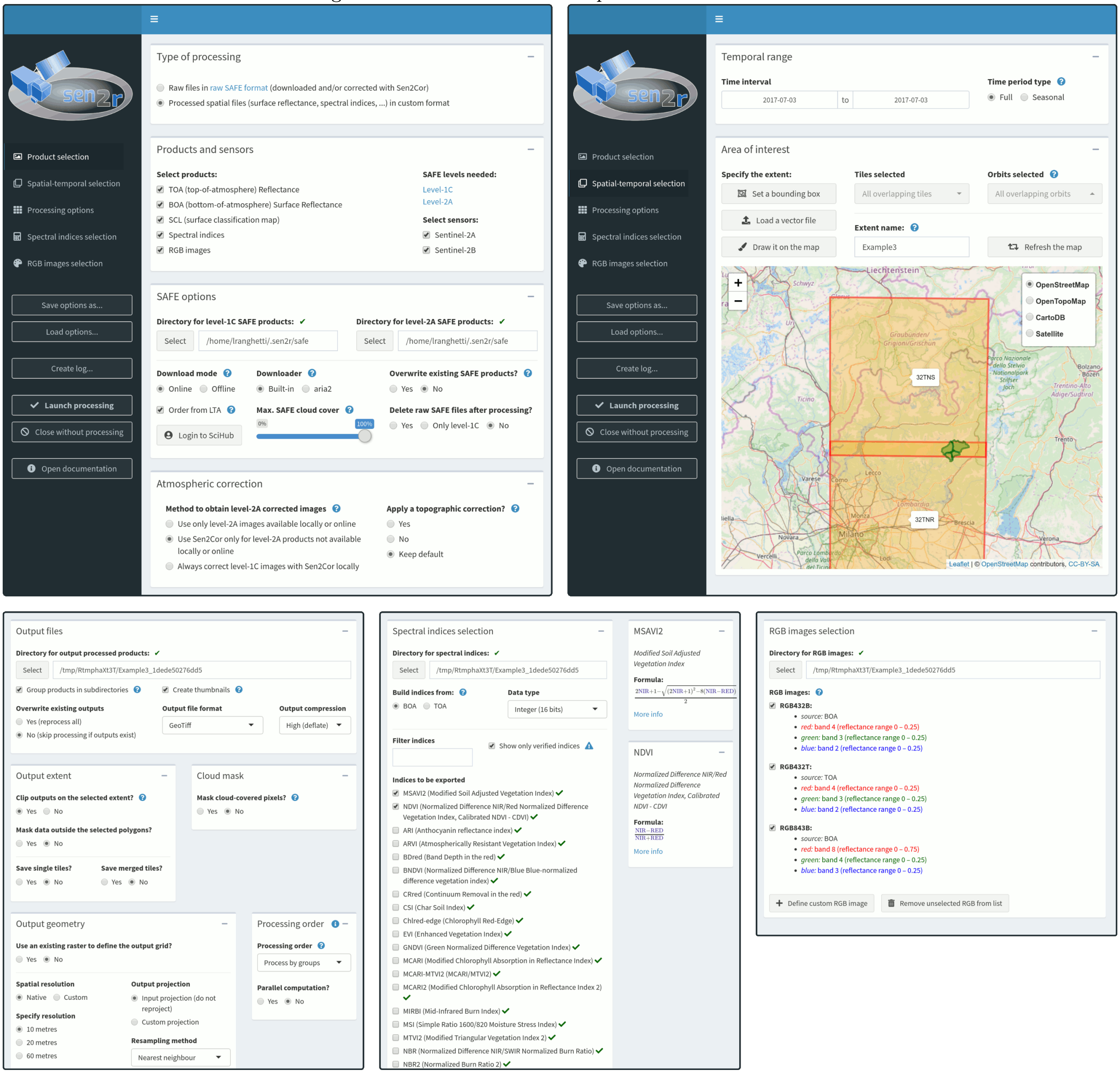
Output extent. This box determines how the AOI defined in the previous panel has to be used. If "Clip outputs on the selected extent" is set to "No", the AOI is used only to identify the required S2 tiles and the output extent corresponds to the full extent of the selected tiles; otherwise, the bounding box of the AOI is used to clip the images. Pixels outside the polygon(s) representing the AOI are either kept or masked (set to NA) based on settings of the corresponding selector. Finally, users can decide if temporary files corresponding to whole tiles and mosaics (see figure 1) should be maintained.

Cloud mask. This box allows deciding if a cloud mask based on the SCL layer should be applied to the output images (see section 2.4 for further details on how cloud masking is performed). If so, a default set of possible masks (based on different combinations of SCL classes) is proposed; alternatively, users can define a custom combination of SCL classes to be masked.

Other settings allow changing the maximum cloud cover detected over the AOI above which output products are not generated, and modifying parameters related to buffering/smoothing of the mask layer (see section 2.4).

Output geometry. This box allows setting the desired spatial resolution and projection of the outputs. Users can either specify the parameters manually, or select an existing raster file to define the output grid (this ensures that outputs are perfectly aligned with the selected template raster). The resampling method used in warping operations can also be specified.

Figure 2: Screenshot of the five panels of the sen2r GUI.



Processing order. This box allows defining which of the four processing schemes described in section 2.8 should be used: "by groups" (default), "by date", "mixed" and "step by step".

Additionally, users can set if parallel computation should be performed; if so, the maximum number of CPU cores to use can be manually defined or automatically determined (in the latter case, it would correspond to the number of available cores minus 2, up to a maximum of 8).

3.2.4. Spectral indices selection

This panel allows selecting the SIs to be computed from a predefined list, decide if they should be computed from TOA or (when available) BOA reflectances, and set the folder where they are saved. When an index is selected, the corresponding formula is shown on the right. Available indices were derived from the IDB data base of remote sensing indices (Henrich et al., 2018, 2009); The correctness of formulae of some of the most commonly used SIs has been manually verified and eventually corrected. By default, only these SIs included are shown in the GUI [10].

3.2.5. RGB images selection

This panel allows specifying the list of RGB images to be generated (see section 2.6). Clicking the "Define custom RGB image" opens a dialog box which allows setting: i) the data source (BOA or TOA); ii) the bands to be used for

[10] A more extensive list of SIs can be visualised by unchecking the option "show only verified indices"; users are however warned to carefully verify correctness of the formulae before computing non-verified indices.

the red, green and blue channels; iii) the reflectance range used for stretching each band.

3.3. Non-interactive execution

Although the easiest way to launch a complete processing chain is by setting parameters with the GUI and launching it right away, it is often useful to be able to launch a processing from the command line without opening the GUI. This allows using sen2r functionalities (provided both by the main function `sen2r()` – see section 3.3.1 – or by other package functions – see section 3.3.2) as part of more complex scripts, or scheduling a processing so to automatically update a time series of S2 products – see section 3.3.3.

3.3.1. Using function `sen2r()` non-interactively

Function `sen2r()` can be used also in non-interactive mode.

Users can set the desired parameters with the GUI, export them to a JSON file and run the command `sen2r()` specifying the JSON path in the argument `param_list`, eventually modifying some of the processing options.

For example, starting from an existing JSON parameter file [11]:

```
#### Non-interactive example 1 ####
## Set the path of an existing SAFE
# param_path_1 <- "/tmp/saved_param.json"
## (for testing: create a sample JSON)
param_path_1 <- build_example_param_file()
## Run sen2r() function
sen2r(param_list = param_path_1)
```

The last command launches the processing chain without opening the GUI, reading the parameters from the options file defined in variable `param_path_1`.

It is also possible to pass all or only specific parameters to `sen2r()` using function arguments [12]. This can be useful in several situations, such as when users need to produce a standalone R script (which does not point to any external JSON path), or to be able to use R objects as processing parameters (e.g. a spatial object as AOI), as shown in the code chunk below:

```
#### Non-interactive example 2 ####
## Load a spatial file
sample_extent_2 <- system.file(
  "extdata/vector/scalve.kml",
  package = "sen2r")
## Define input / output directories
out_folder_2 <- tempfile(pattern = "Example2_")
safe_folder <-  tempfile(pattern = "safe_")
## Run sen2r() function
sen2r(
  gui = FALSE,
  extent = sample_extent_2,
  extent_name = "Example2",
  timewindow = c("2017-07-01", "2017-07-10"),
  list_prods = "BOA",
  list_indices = "NDVI",
  mask_type = "cloud_medium_proba",
  max_mask = 80,
  path_l1c = safe_folder,
  path_l2a = safe_folder,
  path_out = out_folder_2
)
```

This example computes the BOA and NDVI time series from date 2017-07-01 to 2017-07-10, produced over the study area defined in file `"scalve.kml"` and applying a cloud mask.

In addition, the possibility to specify an existing parameter file and modify some of the function arguments allows easily setting-up and run several similar processing chains. As an example, the command:

```
#### Non-interactive example 3 ####
## Define output directory
out_folder_3 <- tempfile(pattern = "Example3_")
## Run sen2r() function
sen2r(
  param_list = param_path_1, # from example 1
  extent = sample_extent_2,  # from example 2
  extent_name = "Example3",
  path_out = out_folder_3
)
```

would replicate the processing chain defined in JSON file in `param_path_1` path on a different AOI (defined with the spatial object `sample_extent_2`), saving the output products in a dedicated directory.

3.3.2. Other R functions

Additional exported sen2r functions can be used to autonomously perform specific intermediate processing steps; table 2 reports the main ones. Accessory functions (table 3) allow also to retrieve ancillary information such as the dates of satellite overpass over a specific AOI, the footprints of S2 tiles, and various metadata obtainable from S2 SAFE archives / filenames or from sen2r products. The complete list of exported functions, along with detailed information and reproducible examples, can be found at `http://sen2r.ranghetti.info/reference`.

3.3.3. Scheduled processing

The non-interactive `sen2r()` mode can be exploited to update archives of output products in near real-time as

[11] For reproducibility reasons, the instruction to set the path to an existing JSON was commented and replaced by a command that generates a sample parameter file.

[12] See `http://sen2r.ranghetti.info/reference/sen2r` for a detailed explanation of argument names and corresponding valid and default values.

Table 2: R functions available in sen2r to perform intermediate processing steps.

Function	Description
`s2_list()`	Retrieve from SciHub a list of S2 products available over an AOI in a time window.
`s2_download()`	Download a list of S2 products.
`s2_order()`	Order a list of S2 products from the Long Term Archive.
`sen2cor()`	Correct L1C products using Sen2Cor.
`s2_mask()`	Create and apply cloud masks based on SCL.
`s2_rgb()`	Create RGB images from S2 reflectances.
`s2_calcindices()`	Compute SIs from S2 reflectances.
`s2_thumbnails()`	Create thumbnails from sen2r products.

Table 3: R accessory functions available in sen2r to manage Sentinel-2 products.

Function	Description
`safe_getMetadata()`	Get information from SAFE filenames or existing S2 archives.
`sen2r_getElements()`	Get metadata from images produced by sen2r.
`safe_is_online()`	Check which S2 archives are available for direct download.
`s2_dop()`	Return the Dates Of Passage of Sentinel-2 satellites over orbits.
`s2_tiles()`	Return Sentinel-2 tiles footprints as a 'R' spatial object.
`list_indices()`	Return information about supported SIs.
`build_example_param_file()`	Build an example JSON parameter file (for testing and code reproducibility).

soon as a new S2 image is made available on SciHub, using job schedulers available on various operating systems, like Cron and systemd on Linux systems or the Windows Task Scheduler on Windows.

Two examples of using Cron to schedule a processing chain on Linux are shown below:

```
30 2 * * * /usr/bin/R -e "sen2r::sen2r('/tmp/saved_param.json')"
 0 4 * * * /usr/bin/Rscript /path/of/script.R
```

In the first case, all processing operations are managed by the instruction `sen2r('/tmp/saved_param.json')`, scheduled every day at 2:30 AM; in the second case, the R script "`/path/of/script.R`", including `sen2r()` within a more complex processing, is scheduled at 4:00 AM.

3.4. Output products

3.4.1. Output format and naming convention

Output images are named based on the following schema:

`S2mll_date_orb_aoi_prod_res.ext`

(e.g. "`S2B2A_20180805_065_Scalve_MSAVI2_10.tif`")

where **`S2mll`** (length: 5) identifies the mission ID ("`S2A`" or "`S2B`") and product level ("`1C`" or "`2A`"); **`date`** (length: 8) is the acquisition date (2018-08-05 in the example above); **`orb`** (length: 3) is the orbit number; **`aoi`** is a alphanumeric string specified by the user to describe the AOI (see section 2.3); **`prod`** is an abbreviation definying the output product type (BOA, TOA, SCL, name of a SI [13], or describing an RGB image [14]); **`res`** (length: 2) is the minimum spatial resolution in metres of the original S2 bands used to generate the product (10, 20 or 60); **`ext`** is the file extension.

3.4.2. Benchmarking

sen2r processing speed is affected by several factors related to processing options (e.g., spatial and temporal extents, type and number of required output products, cloud mask settings) and system performance (e.g., number of CPU cores and download speed).

To provide a benchmark of sen2r performance, a typical use case was therefore tested on a standard PC desktop, a workstation and a cloud infrastructure [15]. The con-

[13] The list of supported index names can be obtained with the function `list_indices("name")`.

[14] RGB image names are 7-length strings with the following structure: `RGBrgbX` (e.g. `RGBb84B`), where **`r`**, **`g`** and **`b`** are the number of the bands to be used respectively for red, green and blue, in hexadecimal format (bands 11, 8 and 4 in the example above), and **`X`** is "`B`" if source is BOA or "`T`" if source is TOA.

[15] **PC Desktop:** *OS:* ArchLinux (Linux version 5.2.13-64); *RAM:* 16 GB; *CPU:* 6 processors Intel(R) Core(TM) i5-8400 2.80GHz; *download speed:* 95 Mbit/s; *read/write speed:* 155 MB/s, 110 MB/s. **Workstation:** *OS:* Ubuntu 18.04.3 LTS (Linux version 4.15.0-64); *RAM:* 128 GB; *CPU:* 32 processors Intel(R) Xeon(R) CPU E5-2650 2.00GHz; *download speed:* 270 Mbit/s; *read/write speed:* 120 MB/s, 305 MB/s. **Cloud virtual machine:** *OS:* Ubuntu 18.04.3 LTS (Linux version 4.15.0-65); *RAM:* 32 GB; *CPU:* 16 processors Intel(R)

sidered use case consists in processing one year of images over an extent of 19×15 km, with six required output products: BOA reflectances, SCL maps, two spectral indices and two RGB composite images, exploiting parallel processing (the default processing scheme is used). The processing involves downloading level 2A SAFE products, merging two adjacent tiles for each date, clipping over the AOI boundaries, applying a cloud mask and creating the required products [16]. Processing requires downloading 286 SAFE archives, totalling about 192 GB. Output consists of 714 raster files (plus as many thumbnails): 95 images per masked products (spectral indices) and 143 images per non-masked ones (SCL maps and RGB).

On the desktop PC used to perform the test, the whole operations required 7 hours 13 minutes to be concluded, of which 6 hours 33 minutes (91%) to download SAFE archives. On the workstation and the cloud infrastructure, time reduced respectively to 4 hours 58 minutes and 1 hours 41 minutes (of which 4 hours 39 minutes – 94% – and 1 hour 16 minutes – 75% – for SAFE download).

4. Using sen2r as processing back-end: a case study for precision farming applications

sen2r functionalities were exploited in 2018 in the framework of the SATURNO project (`https://www.progettosaturno.it`). The project wants to demonstrate the possibility of exploiting near real time S2 satellite data together with crop modelling and smart technologies to implement more sustainable fertilisation practices in rice farming systems.

In this framework, S2 data were used to provide phenological information on crop growth and within field variability of crop conditions. The 5-day S2 acquisition frequency is in fact adequate to create sufficiently dense time series for agronomic monitoring purposes, while its decametric spatial resolution allows discriminating the spatial variability at inter- or intra-field scale.

Within the project, sen2r was therefore used as the processing back-end of the SATURNO geoportal, to generate products used by both remote sensing experts and final users of EO products. Requirements were i) to compute Sentinel-2 SIs and RGB images over the entire study area (Lomellina – PV – Italy), ii) to automatically keep the archive updated as soon as new images are made available, iii) to make EO products readily available to the end users on the geoportal of the project.

Points i) and ii) were managed using a cron execution of a `sen2r()` processing chain, which every day performed the following steps: search and download for new S2 level 2A images overlapping the AOI (three S2 tiles – 32TMQ, 32TMR and 32TNR), acquired in orbit 65; merge and clip them over the extent of the *Lomellina* rice district; apply a cloud mask [17]; produce RGB true colour images and compute the NDRE (Normalised Difference Red Edge) index.

New products were automatically and immediately uploaded to a GeoServer infrastructure, and provided to end users thanks to the GET-IT software suite (Lanucara et al., 2017). In particular, single-date NDRE maps were immediately made available once produced and shown using a colour table aimed at facilitating interpretation by farmers (see figure 3). NDRE images were also automatically ingested in a dedicated processing chain to generate subfield management unit zones for precision agriculture applications. In particular, these products were exploited in smart scouting procedures to support the acquisition of field data by operators, and to create prescription maps for nitrogen fertilisation on the basis of such data and expert knowledge.

A different web interface application was built to analyse temporal profiles of SIs, aggregated on the basis of sub-field zones defined by the prescription maps used for fertilisation. This allowed operators to investigate crop growth variability related to variable fertilisation, in relation with rice phenological stages, estimated through the WARM crop modelling system (Pagani et al., 2019; Gilardelli et al., 2019) (see figure 4). This information is very useful for monitoring the effect of variable rate fertilisation strategies during the crop season, and assessing their effectiveness at the end of it. For example, in Figure 4 NDRE time series clearly show the behaviour of each zone in relation to field average (grey dotted line). Based on the observed behaviour, the sub-fields zones showing higher NDRE at tillering and panicle initiation (yellow, orange and red lines) were assumed to be in better conditions, and therefore received less Nitrogen, allowing the reduction of the total amount of fertiliser applied on the parcel. Conversely, the zones with lower NDRE, assumed to be in worse conditions, received more Nitrogen, so to grant them a further biomass increase before harvesting. A reduction in the NDRE variation at the end of the season among different zones can be interpreted as an effectiveness of the variable-rate fertilisation.

5. Discussion

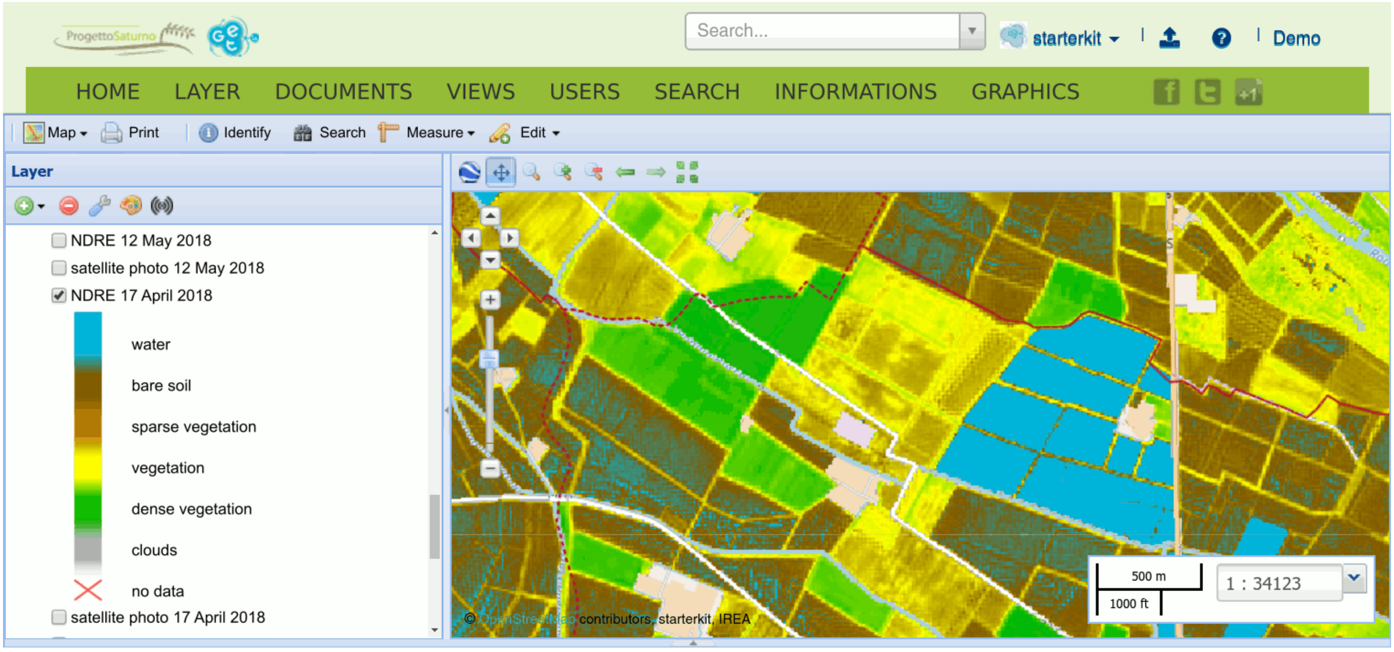
Based on the description of the structure and functionalities of the package described in the previous sections, it can be appreciated that sen2r main aim is not to provide algorithms to perform a complete range of satellite image analysis methods – for which specific applications exist (e.g. the aforementioned Sentinel-2 Toolbox, or dedicated image analysis software (e.g., Orfeo Toolbox, ENVI, ERDAS Imaging) – but rather to manage common prepro-

Xeon(R) CPU E5-2690 v4 2.60GHz; *download speed:* 870 Mbit/s; *read/write speed:* 625 MB/s, 875 MB/s.

[16] The processing chain used for the test can be reproduced using code available at `https://gist.github.com/ranghetti/c0ea41182adb402c44877a263eaf500b`.

[17] Cloud masking was performed using the SCL classes "no data", "cloud medium probability", "cloud high probability" and "snow", and applying a smooth factor of 20 m and a buffer of 150 m to the resulting layer.

Figure 3: SATURNO Screenshot of the GET-IT geoportal, showing a detail of an NDRE image (generated by sen2r) over agricultural parcels with different crop types. The labels in the web interface were translated in English from the original Italian version (accessible at http://saturno.get-it.it/maps/185/view). The colour scale used for NDRE allows easily interpreting land conditions (NDRE < 0.05 was interpreted as "water", 0.05 to 0.15 as "bare soil", 0.15 to 0.60 as "sparse vegetation", 0.60 to 0.90 as "vegetation", 0.90 to 1 as "dense vegetation").



cessing operations, thus facilitating the creation of Analysis Ready Data (ARD) for Sentinel-2 time series. In particular, it allows the execution of a typical processing chain on S2 data, including download, atmospheric correction, geometric transformation (subsetting, masking, reprojecting), cloud masking, computation of spectral indices and production of true / false colour RGB images. A powerful and user friendly GUI is available: far from being a replacement for the standard command line interface to package functions, it allows users to easily set processing parameters and save them as a JSON text file.

The possibility to launch a non-interactive `sen2r()` execution exploiting a previously-saved parameter file is particularly useful for scheduling a processing chain allowing users to periodically update their data archives when new images are made available by the ESA. Advanced R users can also take advantage of the availability of intermediate processing functions, providing powerful instruments to build custom scripts for S2 data analysis. Finally, the possibility to exploit different processing schemes (in terms of order of execution of `sen2r()` internal procedures and parallelisation, as described in section 2.8), is useful to tweak sen2r speed of execution and RAM and disk storage requirements in order to fit users' needs and hardware infrastructures.

As testified by the benchmarks reported in section 3.4.2, the tool can be efficiently used both on standard desktop machines and on powerful server infrastructures. On the reported test case, the limiting factor influencing processing time is indeed the download of SAFE archives, this underlining the efficiency of sen2r functions even on low specification machines. Testing the package on a cloud infrastructure showed a great speed improvement both in terms of downloading and processing time, demonstrates the potentiality of the package in terms of scalability.

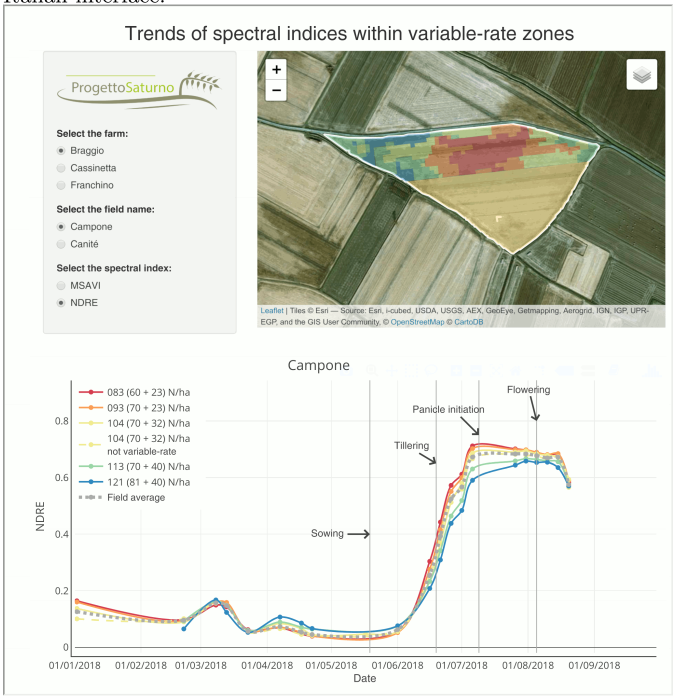
6. Conclusions

This paper presented the new R package sen2r, developed to provide an open-source tool to simplify the preprocessing of Sentinel-2 imagery.

In consideration of the particular characteristics of S2 data (high temporal frequency, decametric resolution and free access), the characteristics of sen2r summarised in section 5 makes it in our opinion a very useful tool for a widespread audience of Sentinel-2 data users. It will facilitate researchers with limited programming skills interested in easily accessing and processing a limited set of images, data scientists interested in analysis of larger spatial/temporal extents, or software engineers interested in deploying web applications allowing end-users to visualise and/or process S2 time series. The case study described in this manuscript demonstrates the usefulness of the package in an operational framework: the scheduled execution of a single function allowed the management of the whole processing chain required to populate the geoportal of the SATURNO project.

Although the package is already fully functional, further development is scheduled in order to implement additional useful functions, such as the possibility to ap-

Figure 4: Web application, accessible at `http://saturno.get-it.it/buttetin`, showing the time series of NDRE generated with sen2r over some rice fields used as test sites in the framework of the SATURNO project. The image was translated in English from the original Italian interface.



ply different atmospheric correction algorithms to L1C data like MAJA (`https://github.com/CNES/Start-MAJA`), ACOLITE (`https://github.com/acolite/acolite`) or C2RCC (`https://github.com/bcdev/s3tbx-c2rcc`) instead of Sen2Cor, or to use S2 level 2A data produced by the CNES MUSCATE production centre (THEIA Land data centre, 2019) with the MAJA processor. Moreover, exploitation of alternative R packages for raster geoprocessing – currently in various development stages – like stars (Pebesma, 2019), gdalcubes (Appel and Pebesma, 2019) and terra (`https://github.com/rspatial/terra`) within sen2r will be evaluated in order to improve the processing performance and simplify the code base.

Acknowledgements

The authors are grateful to the developers of the various R packages on which sen2r depends and to the European Union's Copernicus Programme for acquiring and distributing Sentinel-2 imagery.

The authors thank one of the anonymous reviewers for his his constructive comments, which contributed significantly to improve the quality of the paper.

Activities were funded using residual funds (*"residui"*) of the CNR DTA.AD004.037 (ERMES FP7 – EU grant agreement 606983) and by SATURNO project founded by *Regione Lombardia* FEASR - PSR 2014-2020 program (*Programma di Sviluppo Rurale Misura 1 – Sottomisura 1.2 – Operazione 1.2.01*).

Computer code availability

sen2r is released under the GNU General Public License version 3 (GPL-3); source code is available at `https://github.com/ranghetti/sen2r`. Latest package version is v. 1.3.2 (Ranghetti and Busetto, 2019).

Abbreviations

The following abbreviations are used in this manuscript.

AOI Area Of Interest, being the spatial extension considered in a specific sen2r processing chain

ARD Analysis Ready Data

BOA Bottom Of Atmosphere surface reflectances

GUI Graphical User Interface, referring to the sen2r Shiny interface (see section 3.2)

L1C level 1C Sentinel-2 products, containing TOA reflectances data (see `https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types/level-1c`)

L2A level 2A Sentinel-2 products, containing BOA reflectances and the SCL layer (see `https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types/level-2a`)

LTA Long Term Archive, used to store old S2 archives.

RGB Red-Green-Blue, referring to 3-bands images (see section 2.6)

S2 Sentinel-2

SAFE Standard Archive Format for Europe, the file format used by ESA Earth Observation archiving facilities to make Sentinel-2 data available (see `https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/data-formats`)

SCL Scene Classification Map, an additional output of the Sen2Cor processor included in L2A products (see `https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels/level-2`)

SI Spectral Index

SR Surface Reflectance, referring generically to both BOA and TOA products

TOA Top Of Atmosphere surface reflectances

References

Allaire, J., Ushey, K., Tang, Y., Eddelbuettel, D., Lewis, B., Geelnard, M., 2018. reticulate: interface to 'Python'. URL: https://rstudio.github.io/reticulate. r package version 1.10.

Appel, M., Pebesma, E., 2019. On-demand processing of data cubes from satellite image collections with the gdalcubes library. Data 4. URL: https://www.mdpi.com/2306-5729/4/3/92, doi:10.3390/data4030092.

Appelhans, T., Russell, K., 2018. mapedit: Interactive Editing of Spatial Data in R. URL: https://CRAN.R-project.org/package=mapedit. r package version 0.4.3.

Attali, D., 2018. shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds. URL: https://CRAN.R-project.org/package=shinyjs. r package version 1.0.

Bache, S.M., Wickham, H., 2014. magrittr: A Forward-Pipe Operator for R. URL: https://CRAN.R-project.org/package=magrittr. r package version 1.5.

Baetens, L., Desjardins, C., Hagolle, O., 2019. Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure. Remote Sensing 11, 433. URL: http://www.mdpi.com/2072-4292/11/4/433, doi:10.3390/rs11040433.

Belgiu, M., Csillik, O., 2018. Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis. Remote Sensing of Environment 204, 509 – 523. URL: http://www.sciencedirect.com/science/article/pii/S0034425717304686, doi:https://doi.org/10.1016/j.rse.2017.10.005.

Bivand, R., 2019. CRAN Task View: Analysis of Spatial Data. https://cran.r-project.org/view=Spatial. (version 2019-06-08, accessed 2019-10-14).

Boschetti, M., Busetto, L., Ranghetti, L., Garcia-Haro, J., Campos-Taberner, M., Confalonieri, R., 2018. Testing multi-sensors time series of lai estimates to monitor rice phenology: Preliminary results, in: IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 8221–8224. URL: https://ieeexplore.ieee.org/abstract/document/8518494, doi:10.1109/IGARSS.2018.8518494.

Bruce, L.M., Mathur, A., Jr., J.D.B., 2006. Denoising and wavelet-based feature extraction of modis multi-temporal vegetation signatures. GIScience & Remote Sensing 43, 67–77. URL: https://doi.org/10.2747/1548-1603.43.1.67, doi:10.2747/1548-1603.43.1.67.

Busetto, L., Casteleyn, S., Granell, C., Pepe, M., Barbieri, M., Campos-Taberner, M., Casa, R., Collivignarelli, F., Confalonieri, R., Crema, A., Garcia-Haro, F., Gatti, L., Gitas, I., Gonzalez-Perez, A., Grau-Muedra, G., Guarneri, T., Holecz, F., Katsantonis, D., Minakou, C., Miralles, I., Movedi, E., Nutini, F., Pagani, V., Palombo, A., Paola, F., Pascucci, S., Pignatti, S., Rampini, A., Ranghetti, L., Ricciardelli, E., Romano, F., Stavrakoudis, D., Stroppiana, D., Viggiano, M., Boschetti, M., 2017. Downstream services for rice crop monitoring in europe: From regional to local scale. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10, 5423–5441. URL: https://ieeexplore.ieee.org/document/7898821, doi:10.1109/JSTARS.2017.2679159.

Campos-Taberner, M., García-Haro, F.J., Camps-Valls, G., Grau-Muedra, G., Nutini, F., Busetto, L., Katsantonis, D., Stavrakoudis, D., Minakou, C., Gatti, L., Barbieri, M., Holecz, F., Stroppiana, D., Boschetti, M., 2017. Exploitation of sar and optical sentinel data to detect rice crop and estimate seasonal dynamics of leaf area index. Remote Sensing 9, 248. URL: http://www.mdpi.com/2072-4292/9/3/248, doi:10.3390/rs9030248.

Campos-Taberner, M., García-Haro, F.J., Camps-Valls, G., Grau-Muedra, G., Nutini, F., Crema, A., Boschetti, M., 2016. Multitemporal and multiresolution leaf area index retrieval for operational local rice crop monitoring. Remote Sensing of Environment 187, 102 – 118. URL: http://www.sciencedirect.com/science/article/pii/S0034425716303819, doi:https://doi.org/10.1016/j.rse.2016.10.009.

Chamberlain, S., Teucher, A., 2018. geojsonio: Convert Data from and to 'GeoJSON' or 'TopoJSON'. URL: https://CRAN.R-project.org/package=geojsonio. r package version 0.6.0.

Chang, W., Borges Ribeiro, B., 2018. shinydashboard: Create Dashboards with 'Shiny'. URL: https://CRAN.R-project.org/package=shinydashboard. r package version 0.7.1.

Chang, W., Cheng, J., Allaire, J., Xie, Y., McPherson, J., 2018. shiny: Web Application Framework for R. URL: https://CRAN.R-project.org/package=shiny. r package version 1.2.0.

Cheng, J., Karambelkar, B., Xie, Y., 2018. leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library. URL: https://CRAN.R-project.org/package=leaflet. r package version 2.0.2.

Clauss, K., 2019. Awesome sentinel. a curated list of awesome tools, tutorials and apis related to data from the copernicus sentinel satellites. https://github.com/Fernerkundung/awesome-sentinel. Accessed: 2019-02-05.

Dowle, M., Srinivasan, A., 2019. data.table: Extension of 'data.frame'. URL: https://CRAN.R-project.org/package=data.table. r package version 1.12.0.

Drusch, M., Bello, U.D., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., Bargellini, P., 2012. Sentinel-2: Esa's optical high-resolution mission for gmes operational services, in: The Sentinel Missions - New Opportunities for Science, pp. 25 – 36. URL: http://www.sciencedirect.com/science/article/pii/S0034425712000636, doi:https://doi.org/10.1016/j.rse.2011.11.026.

Eddelbuettel, D., 2018. digest: Create Compact Hash Digests of R Objects. URL: https://CRAN.R-project.org/package=digest. r package version 0.6.18.

European Space Agency, 2015. SENTINEL-2 User Handbook. URL: https://earth.esa.int/documents/247904/685211/Sentinel-2_User_Handbook.

European Space Agency, 2018a. Sentinel-2 Agriculture. Software User Manual. URL: http://www.esa-sen2agri.org/wp-content/uploads/Sen2Agri_SUM_v2.8.pdf. version 2.8; source code: https://github.com/Sen2Agri/Sen2Agri-System.

European Space Agency, 2018b. Sentinel-2 Toolbox. URL: https://sentinel.esa.int/web/sentinel/toolboxes/sentinel-2. source code: https://github.com/senbox-org/s2tbx.

GDAL/OGR contributors, 2018. GDAL/OGR Geospatial Data Abstraction software Library. Open Source Geospatial Foundation. URL: http://gdal.org.

Gilardelli, C., Stella, T., Confalonieri, R., Ranghetti, L., Campos-Taberner, M., Garcia-Haro, F.J., Boschetti, M., 2019. Downscaling rice yield simulation at sub-field scale using remotely sensed lai data. European Journal of Agronomy 103, 108 – 116. URL: http://www.sciencedirect.com/science/article/pii/S116103011830738X, doi:https://doi.org/10.1016/j.eja.2018.12.003.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google earth engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment 202, 18 – 27. doi:10.1016/j.rse.2017.06.031. big Remotely Sensed Data: tools, applications and experiences.

Hagolle, O., Huc, M., Pascual, D.V., Dedieu, G., 2010. A multi-temporal method for cloud detection, applied to formosat-2, venµs, landsat and sentinel-2 images. Remote Sensing of Environment 114, 1747 – 1755. URL: http://www.sciencedirect.com/science/article/pii/S0034425710000908, doi:https://doi.org/10.1016/j.rse.2010.03.002.

Hansen, M.C., Roy, D.P., Lindquist, E., Adusei, B., Justice, C.O., Altstatt, A., 2008. A method for integrating modis and landsat data for systematic monitoring of forest cover and change in the congo basin. Remote Sensing of Environment 112, 2495 – 2513. URL: http://www.sciencedirect.com/science/article/pii/S0034425707004774, doi:https://doi.org/10.1016/j.rse.2007.11.012. earth Observations for Terrestrial Biodiversity and Ecosystems Special Issue.

Henrich, V., Götze, E., Jung, A., Sandow, C., Thürkow, D., Gläßer,

C., 2009. Development of an online indices database: Motivation, concept and implementation, in: EARSeL proceedings, EARSeL, Tel Aviv, pp. –.
Henrich, V., Krauss, G., Götze, C., Sandow, C., 2018. Index database. a database for remote sensing indices. https://www.indexdatabase.de. Accessed: 2018-11-26.
Hijmans, R.J., 2019. raster: Geographic Data Analysis and Modeling. URL: https://CRAN.R-project.org/package=raster. r package version 2.8-19.
Huang, H., Roy, D.P., Boschetti, L., Zhang, H.K., Yan, L., Kumar, S.S., Gomez-Dans, J., Li, J., 2016. Separability analysis of sentinel-2a multi-spectral instrument (msi) data for burned area discrimination. Remote Sensing 8, 873. URL: http://www.mdpi.com/2072-4292/8/10/873, doi:10.3390/rs8100873.
Immitzer, M., Vuolo, F., Atzberger, C., 2016. First experience with sentinel-2 data for crop and tree species classifications in central europe. Remote Sensing 8, 166. URL: http://www.mdpi.com/2072-4292/8/3/166, doi:10.3390/rs8030166.
Isaacson, B.N., Serbin, S.P., Townsend, P.A., 2012. Detection of relative differences in phenology of forest species using landsat and modis. Landscape Ecology 27, 529–543. URL: https://doi.org/10.1007/s10980-012-9703-x, doi:10.1007/s10980-012-9703-x.
Ju, J., Roy, D.P., 2008. The availability of cloud-free landsat etm+ data over the conterminous united states and globally. Remote Sensing of Environment 112, 1196 – 1211. URL: http://www.sciencedirect.com/science/article/pii/S0034425707004002, doi:https://doi.org/10.1016/j.rse.2007.08.011.
Karambelkar, B., Schloerke, B., 2018. leaflet.extras: Extra Functionality for 'leaflet' Package. URL: https://CRAN.R-project.org/package=leaflet.extras. r package version 1.0.0.
Lang, D.T., the CRAN Team, 2019. XML: Tools for Parsing and Generating XML Within R and S-Plus. URL: https://CRAN.R-project.org/package=XML. r package version 3.98-1.19.
Lanucara, S., Zilioli, M., Oggioni, A., Carrara, P., 2017. Get-it, a software suite for easy, interoperable sharing of ecological data in the long term ecological research network, in: EnviroInfo 2017 "From Science to Society: the Bridge provided by Environmental Informatics", pp. 229–234.
Lefebvre, A., Sannier, C., Corpetti, T., 2016. Monitoring urban areas with sentinel-2a data: Application to the update of the copernicus high resolution layer imperviousness degree. Remote Sensing 8, 606. URL: http://www.mdpi.com/2072-4292/8/7/606, doi:10.3390/rs8070606.
Lovelace, R., Nowosad, J., Muenchow, J., 2019. Geocomputation with R. CRC Press. URL: https://geocompr.robinlovelace.net/.
Main-Knorn, M., Bringfried, P., Louis, J., Debaecker, V., Müller-Wilm, U., Gascon, F., 2017. Sen2cor for sentinel-2, in: Proc.SPIE, p. 1042704. URL: https://doi.org/10.1117/12.2278218, doi:10.1117/12.2278218.
Microsoft Corporation, Weston, S., 2017. foreach: Provides Foreach Looping Construct for R. URL: https://CRAN.R-project.org/package=foreach. r package version 1.4.4.
Microsoft Corporation, Weston, S., 2018. doParallel: Foreach Parallel Adaptor for the 'parallel' Package. URL: https://CRAN.R-project.org/package=doParallel. r package version 1.0.14.
Ooms, J., 2014. The jsonlite package: A practical and consistent mapping between json data and r objects. arXiv:1403.2805 [stat.CO] URL: https://arxiv.org/abs/1403.2805.
Pagani, V., Guarneri, T., Busetto, L., Ranghetti, L., Boschetti, M., Movedi, E., Campos-Taberner, M., Garcia-Haro, F.J., Katsantonis, D., Stavrakoudis, D., Ricciardelli, E., Romano, F., Holecz, F., Collivignarelli, F., Granell, C., Casteleyn, S., Confalonieri, R., 2019. A high-resolution, integrated system for rice yield forecasting at district level. Agricultural Systems 168, 181 – 190. URL: http://www.sciencedirect.com/science/article/pii/S0308521X17305048, doi:https://doi.org/10.1016/j.agsy.2018.05.007.
Paul, F., Winsvold, S.H., Kääb, A., Nagler, T., Schwaizer, G., 2016. Glacier remote sensing using sentinel-2. part ii: Mapping glacier extents and surface facies, and comparison to landsat 8. Remote Sensing 8, 575. URL: http://www.mdpi.com/2072-4292/8/7/575, doi:10.3390/rs8070575.
Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. The R Journal URL: https://journal.r-project.org/archive/2018/RJ-2018-009/index.html.
Pebesma, E., 2019. stars: Spatiotemporal Arrays, Raster and Vector Data Cubes. Https://CRAN.R-project.org/package=stars.
Pedersen, T.L., Nijs, V., Schaffner, T., Nantz, E., 2018. shinyFiles: A Server-Side File System Viewer for Shiny. URL: https://CRAN.R-project.org/package=shinyFiles. r package version 0.7.2.
Perrier, V., Meyer, F., Granjon, D., 2019. shinyWidgets: Custom Inputs Widgets for Shiny. URL: https://CRAN.R-project.org/package=shinyWidgets. r package version 0.4.7.
Pesaresi, M., Corbane, C., Julea, A., Florczyk, A.J., Syrris, V., Soille, P., 2016. Assessment of the added-value of sentinel-2 for detecting built-up areas. Remote Sensing 8, 299. URL: http://www.mdpi.com/2072-4292/8/4/299, doi:10.3390/rs8040299.
R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.
Radoux, J., Chomé, G., Jacques, D.C., Waldner, F., Bellemans, N., Matton, N., Lamarche, C., D'Andrimont, R., Defourny, P., 2016. Sentinel-2's potential for sub-pixel landscape feature detection. Remote Sensing 8, 488. URL: http://www.mdpi.com/2072-4292/8/6/488, doi:10.3390/rs8060488.
Ranghetti, L., Busetto, L., 2019. sen2r: Find, Download and Process Sentinel-2 Data. URL: https://cran.r-project.org/package=sen2r, doi:10.5281/zenodo.3406962. r package version 1.1.0.
Reed, B.C., Schwartz, M.D., Xiao, X., 2009. Remote sensing phenology, in: Noormets, A. (Ed.), Phenology of Ecosystem Processes: Applications in Global Change Research. Springer New York, New York, NY, pp. 231–246. URL: https://doi.org/10.1007/978-1-4419-0026-5_10, doi:10.1007/978-1-4419-0026-5_10.
Richter, R., Louis, J., Müller-Wilm, U., 2012. Sentinel-2 msi–level 2a products algorithm theoretical basis document. European Space Agency,(Special Publication) ESA SP 49, 1–72.
RStudio, Inc, 2013. Easy web applications in R. URL: http://www.rstudio.com/shiny/.
Shi, Y., Huang, W., Ye, H., Ruan, C., Xing, N., Geng, Y., Dong, Y., Peng, D., 2018. Partial least square discriminant analysis based on normalized two-stage vegetation indices for mapping damage from rice diseases using planetscope datasets. Sensors 18, 1901. URL: http://www.mdpi.com/1424-8220/18/6/1901, doi:10.3390/s18061901.
Stendardi, L., Karlsen, S.R., Niedrist, G., Gerdol, R., Zebisch, M., Rossi, M., Notarnicola, C., 2019. Exploiting time series of sentinel-1 and sentinel-2 imagery to detect meadow phenology in mountain regions. Remote Sensing 11. URL: http://www.mdpi.com/2072-4292/11/5/542, doi:10.3390/rs11050542.
THEIA Land data centre, 2019. Copernicus Sentinel-2 data processed at level 2A by CNES. URL: https://theia.cnes.fr/atdistrib/rocket/#/search?collection=SENTINEL2.
Veloso, A., Mermoz, S., Bouvet, A., Toan, T.L., Planells, M., Dejoux, J.F., Ceschia, E., 2017. Understanding the temporal behavior of crops using sentinel-1 and sentinel-2-like data for agricultural applications. Remote Sensing of Environment 199, 415 – 426. URL: http://www.sciencedirect.com/science/article/pii/S0034425717303309, doi:https://doi.org/10.1016/j.rse.2017.07.015.
Verhegghen, A., Eva, H., Ceccherini, G., Achard, F., Gond, V., Gourlet-Fleury, S., Cerutti, P.O., 2016. The potential of sentinel satellites for burnt area mapping and monitoring in the congo basin forests. Remote Sensing 8, 986. URL: http://www.mdpi.com/2072-4292/8/12/986, doi:10.3390/rs8120986.
Vrieling, A., Meroni, M., Darvishzadeh, R., Skidmore, A.K., Wang, T., Zurita-Milla, R., Oosterbeek, K., O'Connor, B., Paganini, M., 2018. Vegetation phenology from sentinel-2 and field cameras for a dutch barrier island. Remote Sensing of Environment 215, 517 – 529. URL: http://www.sciencedirect.com/science/article/pii/S003442571830107X, doi:https://doi.org/10.1016/j.rse.2018.03.014.

Wickham, H., 2018. httr: Tools for Working with URLs and HTTP. URL: https://CRAN.R-project.org/package=httr. r package version 1.4.0.

Zhang, X., Friedl, M.A., Schaaf, C.B., 2009. Sensitivity of vegetation phenology detection to the temporal resolution of satellite data. International Journal of Remote Sensing 30, 2061–2074. URL: https://doi.org/10.1080/01431160802549237, doi:10.1080/01431160802549237.