S2 MPC

Sen2like User Manual

Ref. S2-SEN2LIKE-UM-V1.10

Authors Table

	Name	Company	Responsibility	Date	Signature
Written by	S. Saunier	TPZF	Sen2Like Team	2025-01-30	

Change Log

Issue	Date	Reason for change	Pages(s)/Section(s)
1 Draft A	06 March 2019	Initial Issue.(HLS TPZF S2L UM0017)	All
	26 May 2020	Major update in the CCN context	All
1.1	23 June 2020	Final version	All
1.2	14 Oct. 2020	Update with new features	All
1.3	21 Sept. 2021	Update for release v3.2.0 - Maja product Format specification - To clarify band Parallelisation - Reference LS09	All
1.4	07 Dec. 2021	Update for release dedicated to massive production	All
1.5	17 Jan. 2022	Update for release 4.0	All
1.6	28 Oct 2022	Update for release 4.1 (Leucate) ROI Based mode: - Update §2.1 - Add §2.4.4 - Update §4 - Add §4.1.4 - Add §4.3.4 Mixed Archives: - Update §5.1.2.3 - Add §5.1.2.3.4	All

Issue	Date	Reason for change	Pages(s)/Section(s)
1.7	02 Feb 2023	Update for release 4.2 (Leman): Changes for different UTM management and Geometric Processing and QC - Update Figure 1 - Update §2.2 - Update Table 3 - Update Figure 7 - Update Table 5 (remove UTM limitation) - Update §4 - Update §4.2: add --allow-other-srs argument - Update §5.1.2.1: update processing block config - Update §5.1.2.9: add stitching same_utm_only config parameter - Update $5.2.1 HR BRDF data - Update §5.3: internal processing block config - Update Appendix A: config sample	All
1.8	16 June 2023	Update for release 4.3 (Balaton): - Figure-1 Overview of Sen2Like processing workflow updated - Paragraphs related to DEM downloader removed as its configuration sample - PRISMA mission integration: chapter 2.1 updated and Appendix C created	All, 2.1, Appendix C
1.9		Update for release 4.4 (Victoria): - Figure-1 Add Topographic Correction - Add Topographic Correction description in chapter 2.2 - Update SBAF description in chapter 2.2 - Update Figure 7 - Add chapter 4.4 for DEM Downloader utility - Add new config parameters in chapter 5.1.2 (DEM Repository, Topographic Correction, Sbaf sections) - Add chapter 5.2.5: describe DEM folder tree expected by sen2like - Update Offset coefficient to 1000 to comply with the new S2 format $5.1.2.12	All

Issue	Date	Reason for change	Pages(s)/Section(s)
1.10		Update for Sen2like 4.5: - Update Python version - Update conda environment creation - Update usage - Add missing sen2cor_topographic_correction parameter description - typo	Update for Sen2like 4.5: - Chapter 2.5 - Chapter 3.1 - Chapter 4 - Chapter 5.1.2.10 - All

This Page Is Intentionally Blank

Table of contents

This Page Is Intentionally Blank

1. INTRODUCTION

1.1 Purpose of the document

This document is the user manual document of the "Sen2Like" phase 3 software, developed in the frame of the two following S2 Mission Performance Centre (MPC) Contract Change Requests; Sen2Like phase 1 [ESA-EOPG-Cop-CR-1], Sen2Like phase 2 [ESA-EOPG-Cop-CR-11]; Sen2Like phase 3 [ESA-EOPG-Cop-CR-19] (contract ref. 4000108650/13/I-LG).

1.2 Document structure

The document is structured as follows:

Chapter 1 – This introduction

Chapter 2 – This chapter describes the tool.

Chapter 3 – This chapter provides the installation note of the software package.

Chapter 4 – This chapter details the command lines for proper execution.

Chapter 5 – The chapter addresses all aspects related to the configuration: Processor & Auxiliary data.

1.3 References

The reference list of all projects related documents with their version number and issue date is given in:

[RD.1] Sen2Like, a tool to generate Sentinel-2 Harmonised Surface Reflectance Products, First Results With Landsat-8, 3rd S2 Validation Team Meeting1

[RD.2] S. Skakun, J. Ju, M. Claverie, J.C Roger, E. Vermote, B. Franch, J.L Dungan and J. Masek. Harmonized Landsat Sentinel-2 (HLS) Product User's Guide. Version 1.4, October 2018.[2]

[RD.3] Saunier, S.; Pflug, B.; Lobos, I.M.; Franch, B.; Louis, J.; De Los Reyes, R.; Debaecker, V.; Cadau, E.G.; Boccia, V.; Gascon, F.; Kocaman, S. Sen2Like: Paving the Way towards Harmonization and Fusion of Optical Data. Remote Sens. 2022, 14, 3855. https://doi.org/10.3390/rs14163855 .

[1] https://www.researchgate.net/publication/332428332_Sen2like_a_Tool_to_Generate_Sentinel-2_Harmonised_Surface_Reflectance_Products_-_First_Results_With_Landsat-8

[2] https://hls.gsfc.nasa.gov/wp-content/uploads/2018/10/HLS.v1.4.UserGuide_draft_ver3.0_clean.pdf

[RD.4] Landsat 8 Level 1 Data Format Control Book (DFCB), Version 11.0, February 2017[3]

[RD.5] Landsat 8-9 OLI/TIRS Collection 2 Level 2 Data Format Control Book (DFCB), Version 6.0, September 2020

[RD.6] Sentinel-2 Products Specification Document, Version 14.5, March 2018 [S2-PDGS-TAS-DI-PSD]4

[RD.7] Technical Note, Sentinel-2A L2A Product description, Version 1.0, May 2016[5].

[RD.8] Level 2HF Product Format Specification, Version 1.1, January 2022 [S2-PDGS-MPC-L2HF-PFS].

[RD.9] EOGRID Cloud Tool box https://eogrid.esrin.esa.int/cloudtoolbox/

[RD.10] Sen2Cor Atmospheric corrections tool

[RD.11] https://hls.gsfc.nasa.gov/data/v1.4/ HLS Data

[RD.12] Landsat 8 Quality Reports – https://earth.esa.int/web/sppa/mission-performance/esa-3rd-party-missions/landsat-8/oli-tirs/cyclic-quality-reports

[RD.13] Roy and al. (2017). "Examination of Sentinel-2A multi-spectral instrument (MSI) reflectance anisotropy and the suitability of a general method to normalize MSI reflectance to nadir BRDF adjusted reflectance." Remote Sensing of Environment 199 (2017) 25–38

[RD.14] Vermote, E., C.O. Justice, et F.-M. Breon. « Towards a Generalized Approach for Correction of the BRDF Effect in MODIS Directional Reflectances ». IEEE Transactions on Geoscience and Remote Sensing 47, no 3 (mars 2009): 898 908. https://doi.org/10.1109/TGRS.2008.2005977

[RD.15] Franch, B., Vermote, E., Skakun, S., Roger, J. C., Masek, J., Ju, J., Villaescusa-Nadal, J.L. & Santamaria-Artigas, A. (2019). A method for Landsat and Sentinel-2 (HLS) BRDF normalization. Remote Sensing, 11(6), 632.

[RD.16] Claverie, Martin, Junchang Ju, Jeffrey G. Masek, Jennifer L. Dungan, Eric F. Vermote, Jean-Claude Roger, Sergii V. Skakun, et Christopher Justice. « The Harmonized Landsat and Sentinel-2 Surface Reflectance Data Set ». Remote Sensing of Environment 219 (15 décembre 2018): 145-61. https://doi.org/10.1016/j.rse.2018.09.002.

[RD.17] Gao, F, Masek, J, Schwaller, M, Hall, F (2006). On the blending of the Landsat and MODIS surface reflectance: Predicting daily Landsat surface

3 https://prd-wret.s3.us-west-2.amazonaws.com/assets/palladium/production/atoms/files/LSDS-809-Landsat8-Level1DFCB-v11.pdf

4 https://earth.esa.int/documents/247904/685211/Sentinel-2-Products-Specification-Document

5 https://theia.cnes.fr/atdistrib/documents/PSC-NT-411-0362-CNES_01_00_SENTINEL-2A_L2A_Products_Description.pdf

reflectance, IEEE Transactions on Geoscience and Remote Sensing 44, no 8, 2207-2218

1.4 Informative Reference Documents

[ECSS-E-HB-40A]	Software engineering handbook (11 December 2013), https://ecss.nl/hbstms/ecss-e-hb-40a-software-engineering-handbook-11-december-2013

1.5 Relation to other Documents

There are relation with the following documents:

- [SEN2LIKE-PDD], Sen2Like Output Product Format (PDD)
- [SEN2LIKE-PSD], Sen2Like Output Product Format (PSD)
- [SEN2LIKE-VP], Verification plan
- [SEN2LIKE-SDD], Software Design Document
- [SEN2LIKE-ATBD], Algorithm Theoretical Basis Document
- [SEN2LIKE-ATBD HABA], Algorithm Theoretical Basis Document
- [SEN2LIKE-TR], Test Report

1.6 Definitions of Terms and Conventions

The following acronyms and abbreviations are used in this report.

API	Application Programming Interface
AOI	Area Of Interest
BOA	Bottom Of Atmosphere
BRDF	Bidirectional Reflectance Distribution Function
CAMS	Copernicus Atmosphere Monitoring Service
CESBIO	Center for Space Studies of BIOsphere
CRS	Coordinate Reference System
DEM	Digital Elevation Model
ESA	European Space Agency
EU	European Union
FTP	File Transfer Protocol
KLT	Kanade-Luca-Tomasi
GIPP	Ground Image Processing Parameter
JSON	JavaScript Object Notation
HLS	Harmonized Landsat Sentinel-2
HR	High Resolution
L1	Level 1

L2F	Level 2 Fused (Level-2F)
L2H	Level 2 Harmonized (Level-2H)
MGRS	Military Grid System
MODIS	Moderate Resolution Imaging Spectroradiometer
MPC	Mission Performance Centre
MS	Multi Spectral
MSI	Multi-Spectral Instrument
NASA	National Aeronautics and Space Administration
NBAR	Nadir BRDF-normalized Reflectance
NetCDF	Network Common Data Form
NIR	Near InfraRed
OLI	Operational Land Imager
Pan	Panchromatic
PRISMA	PRecursore IperSpettrale della Missione Applicativa
RD	Reference Document
RGN	Red Green Blue
ROI	Region Of Interest
S2A/S2B	Sentinel-2A / 2B
S2L	Sen2Like
SBAF	Spectral Band Adjustment Factor
SCL	Scene Classification (map)
SMAC	Simplified Model for Atmospheric Corrections
SWIR	Short-Wave InfraRed
TIF	Tagged Image File
URL	Uniform Resource Locator
UTM	Universal Transverse Mercator
VJB	Vermote Justice Breon
WRS	Worldwide Reference System

2. DESCRIPTION

2.1 Main Overview

The Sentinel-2 and Landsat missions have always been of great importance for Earth Observation agricultural applications (Land User / Land Cover) that requires surface reflectance data from Multi-Spectral (MS) High Resolution (HR) instruments.

The scope of Sen2Like is to harmonize Sentinel-2 / Landsat data in order to increase the temporal revisit ([RD.1], [RD.2]) by generating relevant "Sentinel-2 like" products. The "Sen2Like" term refers in fact to the notion of "Sentinel-2" considered as a reference mission as a baseline principle to produce multi temporal data stack.

In this context, the Sen2Like development goals and main features are:

- A data harmonization framework,
- Four production modes: 'Product', 'Single Tile', 'Multi Tile' and 'ROI Based'
- Option to produce in near real time context and in cloud environment (DIAS)
- Delivery of two product types: harmonized ("Level-2H") and fused ("Level-2F") data products

The Level 2F includes Blue, Green, Red Landsat 8 image bands rescaled to 10.0 m pixel spacing.

The Sen2Like software is an open-source solution. Designed as an Earth Observation data demonstration processor, the software ingests Sentinel-2 (Level 1, Level 2)/ Landsat 8 / Landsat 9 (Level 1,level 2) products ([RD.4], [RD.5], [RD.6]), including MAJA products [RD.7].

The software generates, under nominal processing baseline, harmonized product with format described in [RD.8].

In addition to Sentinel-2 and Landsat missions, there have been an effort to integrate the PRISMA Earth Observation mission, which is a medium-resolution hyperspectral imaging satellite, developed, owned and operated by ASI (Agenzia Spaziale Italiana). Sen2like software supports the PRISMA[6] L1 products through the usage of a pre-processor named "prisma4sen2like" that transforms PRISMA L1 products into an internal format: Sentinel-2 PRISMA (S2P) L1C products, spectrally aggregated into 13 Sentinel-2 bands and projected into L1C Sentinel-2 geometry (without refinement on Sentinel-2 GRI). The PRISMA integration approach is described with more details in Appendix C.

As shown in figure below, there are different possible input processing levels depending on the mission. Accordingly, the combination of processing, detailed below, is different. The outputs of the processing are spatio-temporal data stack of Sen2Like products, all image expressed into the Sentinel-2 Military Grid Reference System (MGRS) tiling system.

[6] https://www.asi.it/en/earth-science/prisma/

The output product format can be either Level-2H or Level-2F ([RD.8]). Both formats are quite similar in terms of product structure. The difference is that the Level-2H embeds mission dependant harmonized data and the Level 2F embeds mission independent harmonized data: the resolution of all images from equivalent band is the same by following the Sentinel-2 convention. Whatever the operational mode, the same workflow is followed.

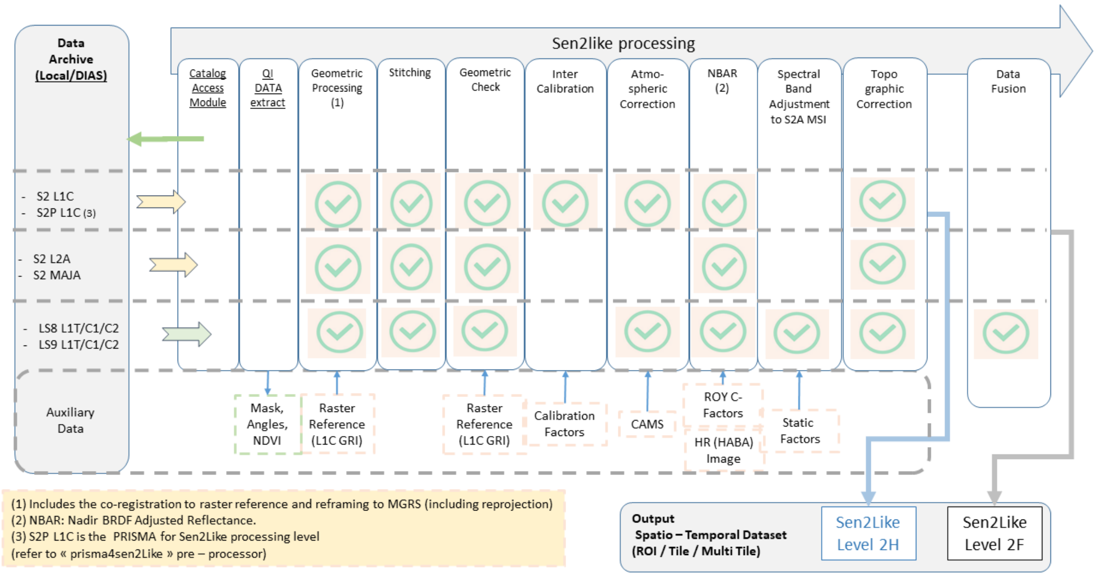

Figure 1 : Overview of Sen2Like processing workflow.

Furthermore, the auxiliary information includes Geometrical Reference data, Digital Elevation Model data (DEM), data from the Copernicus Atmosphere Monitoring Service (CAMS) and geometric reference. Moreover, it optionally includes Bidirectional Reflectance Distribution Function (BRDF) coefficient data and Sentinel-2 L2A Scene Classification map (SCL).

As discussed above, the Sen2Like band designation convention is defined based on the current Sentinel-2 Multi-Spectral Instrument (MSI) band designation whenever possible. The processing is performed for equivalent Sentinel-2 / Landsat 8 spectral channel images. Even if no match exists, the concerned spectral channel images are kept in the final Sen2Like product and stored in a dedicated folder. It is the case for the Landsat 8 thermal / panchromatic data, and for the Sentinel-2 red edge / NIR1 data.

The table below shows the band naming convention adopted in the L2H / L2F products. In addition, for each spectral band, the resolution of the image is provided. The bands for which image fusion algorithm has been applied (L2F) are indicated in bold where corresponding resolution of the image is given. Depending on the band group, the records of the table are displayed with a specific colour. The bands kept as native are indicated in italic. Indeed, mismatch between spectral bands exist; for instance, the thermal bands of Landsat 8 do not get equivalency within the Sentinel-2 product. In this case, band file is considered as "native"; related data is stored within a dedicated 'Native' directory, and no Sen2Like processing is applied.

The band group nomenclature is listed in Table 2 and this table is convenient to describe applicability / validity of each processing, as discussed just hereafter.

Table 1 : Composition of the L2H / L2F products.

Sentinel 2 MSI bands (Center Wavelength [µm])	Lansdat 8 / Landsat 9 bands (Center Wavelength)	Designation	Sen2like Convention	L2H-S2 resolution (m)	L2H-L8 resolution (m)	L2F-S2 resolution (m)	L2F-L8 resolution (m)
B01	B01 (442 nm)	Coastal Aersol	B01	60 m	30 m	60 m	30 m
B02 (490 nm)	B02 (482 nm)	Blue	B02	10 m	**30 m**	10 m	**10 m**
B03 (560nm)	B03 (561 nm)	Green	B03	10 m	**30 m**	10 m	**10 m**
B04 (665 nm)	B04 (654)	Red	B04	10 m	**30 m**	10 m	**10 m**
B08 (842 nm)		*NIR 1*	*B08*	*10 m*	-	*10 m*	-
B8A (865 nm)	B05 (864 nm)	NIR2	B8A	20 m	**30 m**	20 m	**20 m**
B11 (1610 nm)	B06 (1608 nm)	SWIR 1	B11	20 m	**30 m**	20 m	**20 m**
B12 (2190 nm)	B07 (2200 nm)	SWIR 2	B12	20 m	**30 m**	20 m	**20 m**
	B08 (589 nm)	*Panchromatic*	*BP1*	-	*15 m*	-	*15 m*
	B10 (11 µm)	*TIRS 1*	*BT1*	-	*100 m*	-	*100 m*
	B11 (12,2 µm)	*TIRS 2*	*BT2*	-	*100 m*	-	*100 m*
B05 (705 nm)		*Red Edge 1*	*B05*	*20 m*	-	*20 m*	
B06 (740 nm)		*Red Edge 2*	*B06*	*20 m*	-	*20 m*	
B07 (783 nm)		*Red Edge 3*	*B07*	*20 m*	-	*20 m*	

Table 2 : Sen2Like band group convention.

Designation	Band Code Sequence	
Coastal + SWIR	B01, B11,B12	L2F/L2H
RGB	B02,B03,B04	L2F/L2H
NIR	B08,B08A	B8A L2F/L2H B08 "Native"
Pan + Thermal	BP1, BT1, BT2	"Native"
Red Edge	B05,B06,B07	"Native"

For completeness, as given in[7], the definitions of the spectral bands are recalled in the graphic below.

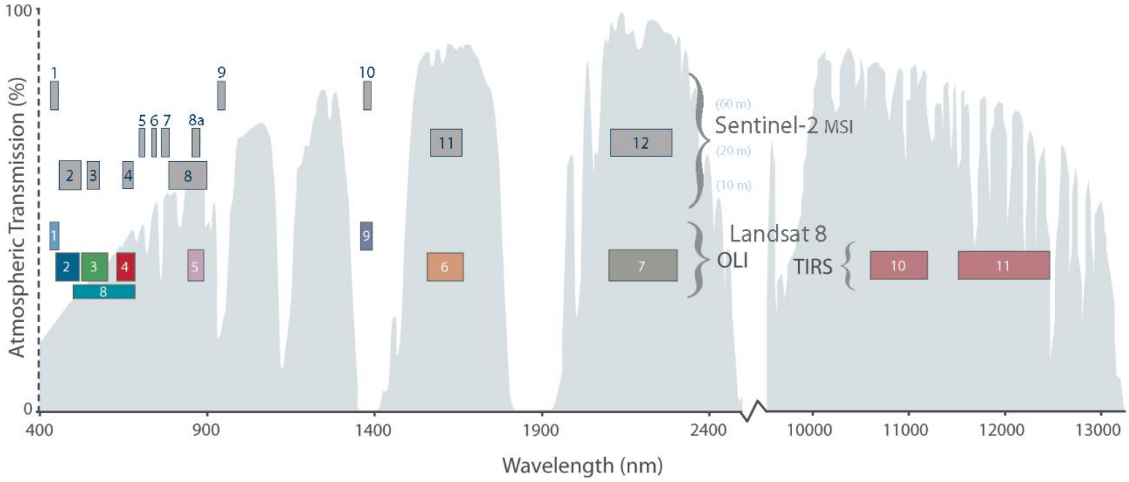

Figure 2 : Sentinel-2 and Landsat 8 spectral band definition.

The Sen2Like prototype processor runs in four following distinct operational modes:

Single tile mode	The processor considers as input one single MGRS tile and a period (start time / end time). By using this mode, all corresponding LS8 (LS9) and S2 data are processed. An MGRS multi temporal stack (L2F / L2H) is finally delivered.
Multi tile mode	The processor considers as input an Area Of Interest (AOI) and a period of time (start time / end time). By using this mode, all MGRS S2 tiles and LS8 (LS9) scenes that overlap the AOI within a selected period are processed. An AOI based multi temporal stack (L2F / L2H) is delivered.
Product mode	The processor considers as input only one LS8 (LS9) or S2 product and apply processing for the corresponding MGRS tile.
ROI based mode	The processor considers as input a Region Of Interest (ROI) and a period of time (start time / end time). By using this mode, all S2 tiles and LS8 (LS9) scenes that fully contain the ROI within a selected period are processed. Only one MGRS tile can be processed. An MGRS multi temporal data stack (L2F / L2H) is finally delivered and the value of pixels outside the ROI extent is set to nodata.

Regarding the location of input data, there are two ways of running Sen2Like. Classic approach consists in using products stored locally. An alternative approach

[7] http://landsat.gsfc.nasa.gov/wp-content/uploads/2015/06/Landsat.v.Sentinel-2.png

consists in using products available from the Creodias infrastructure. In this latter case, the catalogue queries are performed with the CreoDIAS OpenSearch (1.1) like API "Finder"[8]. On the other hand, the data access is done through file system.

2.2 Processing Algorithms Overview

Sen2Like processor performs the following 7 main processing steps:

1. Geometric Processing
2. Stitching
3. Geometric Check
4. Inter-calibration
5. Atmospheric correction
6. Bidirectional Reflectance Distribution Function (BRDF) Adjustment
7. Spectral Band Adjustment Factor (SBAF)
8. Topographic Correction
9. Data Fusion

Geometric Processing

In a systematic way, the input images (all spectral bands) are co-registered to a reference image. The reference image needs to be prepared prior to starting the Sen2Like processing. Note that data from the S2 Global Reference Image (GRI) reference image[9], in the L1C format, can be used as reference, avoiding data preparation stage. As well as geometrically refined S2 products.

The output projection of Sen2Like products is the Sentinel-2 the tiling system. This tiling system is aligned with the UTM-based MGRS[10].

During the geometric processing, both input and reference geometric grids are compared, and co-registration errors estimated. An image matching method based on Kanade-Lucas-Tomasi (KLT) technics[11], [12], [13] is used. The statistics on geolocation errors provide correction factors subsequently applied for geometric co registration of input image.

The co-registration processing by default is forced on S2 products for which geometric refinement has been applied during the L1C processing (As of Processing Baseline 3.00 release date[14]). This mechanism can be anyhow overridden by using appropriate parameter as specified in 5.1.2.4.

It is worth noting that prior to matching, a dedicated framing process clip the Landsat image to fit within the given MGRS tile geographic extent. The Landsat image grid can be expressed into a Coordinate Reference System (CRS) that

[8] https://creodias.eu/eo-data-finder-api-manual

[9] C.Dechoz and al, In proceeding of Living Planet Symposium 2015, Sentinel-2 Global Reference Image

[10] https://en.wikipedia.org/wiki/Military_Grid_Reference_System

[11] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence, pages 674–679, 1981.

[12] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[13] Shi, J.; Tomasi, C. Good Features to Track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600

[14] https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/processing-baseline

differs from the target UTM. In this case, prior to be reframed, the image is re-projected into this target UTM.

Stitching

There are configurations for which several input scenes need to be stitched together to fully covered the geographical extent of a given MGRS Tile. Theses configurations are:

- Two consecutive Landsat Worldwide Reference System (WRS) scenes; the software selects these two scenes for stitching purposes.
- A same MGRS Tile recorded with two distinct data strips[15] (same orbit) that result in two products of the same MGRS tile. These products are selected by the software and image stitching applied.

Within this configuration, the two consecutive Landsat WRS scenes might be expressed into different CRS and harmonized accordingly by using re projection as discussed here before.

Geometry Check

The geometric check process is a Quality Control step of the product (after stitching if any); The same matching process as the one used in the Geometric Processing is played back and the final co-registration accuracy (to reference image) report. The accuracy achieved is within 0.3 pixel (3 sigma).

Inter-calibration

Radiometric validation activities based on different methods and performed by independent teams agree on mis-calibration between Sentinel-2A (S2A) and Sentinel-2B (S2B): S2B VNIR bands being slightly darker with respect to S2A VNIR bands.

Also, the inter-calibration step aims to bring closer the Top of Atmosphere Radiometry of Sentinel-2B (S2B) with the one of Sentinel-2A (S2A). For this purpose, a scaling factor of 1.011 is applied to the following S2B bands: B01, B02, B03, B04, B05, B06, B07, B08, B8A, B09. Nothing is done for S2B SWIR bands (B10, B11, B12).

With the next generation (scheduled in 2022) of S2 L1C and S2 L2A products (processing baseline > 04.00), it is foreseen to have those inter-calibration coefficients already applied during L1 processing[16].

In Sen2like version 3.3 the inter-calibration coefficients are applied to S2B products if this processing is activated (value set to True). It is the responsibility of the user to disable it, in case L1C and L2A products with processing baseline > 04.00 are processed.

The next version of Sen2Like will be able to automatically detect those already inter-calibrated S2B products, in order not to apply the inter-calibration twice.

Atmospheric correction

[15] https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/definitions

[16] https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-2-msi/processing-baseline

Starting from Level-1C, it is possible to perform the conversion from Top Of Atmosphere (TOA) to Bottom Of Atmosphere (BOA) by using the two following methods:

- The Simplified Model for Atmospheric Corrections (SMAC) method[17], based on Look up tables (Sensor coefficients are shared by CESBIO)
- The Sen2Cor method[18], as implemented as part of Sentinel-2 Level-2A baseline production.

Both methods rely on auxiliary data provided by European Centre for Medium-Range Weather Forecasts (EMWF), Copernicus Atmosphere Monitoring Service (CAMS) Near Real Time and Reanalysis data[19]. The Simplified Model for Atmospheric Correction (SMAC) method strongly relies on the statistical processing of auxiliary information to estimate Ozone Content, Water Vapour, Temperature, Aerosol Optical Thickness at the time / earth location of the satellite overpass.

Bidirectional Reflectance Distribution Function (BRDF) Adjustment

The viewing and illumination angles are accounted to provide nadir BRDF-adjusted reflectance (NBAR). For doing this correction two methods are proposed:

- The c-factors approach: a single set of mission/band dependant coefficients are used to estimate a global BRDF shape as function of viewing, illumination angles, as discussed in given in [RD.13].
- The MGRS Tile BRDF database approach: High resolution BRDF shape characterization is used as detailed in [RD.14] and [RD.15].

The images are normalized per pixel. The view angle is set to nadir, and the solar zenith angle is fixed through time but varies for each tile based on latitude as described in [RD.16]. It is worth noting that this processing is not available for Cirrus, water vapor, MSI Red Edge and OLI Thermal and Pan band.

Spectral Band Adjustment Factor (SBAF)

The harmonization requires adjustment of small differences due to spectral response which is specific to each instrument. In the Sen2Like processing, OLI is rescaled to S2A/MSI and S2B/MSI to S2A/MSI. Characterization results obtained with Hyperion data and proposed in the NASA Harmonized Landsat Sentinel-2 (HLS) project are used [RD.2].

In addition, an experimental approach that consists of use adaptative method to some candidate bands. This method applies band specific slope and offset param to NDVI mask to compute a per pixel factor and the apply the factor matrix to the image band. Notice that if NDVI pixel < 0.1, standard SBAF computation is applied even if adaptive is activated.

Topographic Correction (experimental)

[17] Rahman, H., & Dedieu, G. (1994). "SMAC: a simplified method for the atmospheric correction of satellite measurements in the solar spectrum." Remote Sens., 15(1), 123-143

[18] Main-Knorn, Magdalena & Pflug, Bringfried & Louis, Jerome & Debaecker, Vincent & Müller-Wilm, Uwe & Gascon, Ferran. (2017). Sen2Cor for Sentinel-2. 3. 10.1117/12.2278218.

[19] https://apps.ecmwf.int/data-catalogues/cams-reanalysis/

Please notice that the nominal way to perform topographic correction is to use Sen2cor3.

The objective of the Topographic Correction is to correct for the effects of the different sun illumination of the pixels in the image due to the topography, to limit the sun illumination effects in time series of surface reflectance as well as for land classification applications.

Depending on the atmospheric correction method selected (either Sen2Cor or SMAC), it is possible to activate the Sen2Cor terrain correction or an alternative and simpler method based only on the local solar illumination angle. A topographic correction factor is derived from the map of local solar illumination and applied to the input, considering a limiter to avoid over correction for very high solar zenith angle. From a processing point of view this latter method is interesting because it is fast, band independent and could potentially be applied per pixel.

Data Fusion

The fusion process is the main step involved in the production of fused products (L2F processing level) with Landsat 8 image pixel spacing fully aligned on Sentinel-2 image pixel spacing.

In the literature, in most of the cases high temporal revisit of one sensor is combined with high spatial frequency of another similar sensor in order to produce synthetic data high frequencies in time and space ([RD.17]).

Herein the context is slightly different: compared to Landsat 8, the Sentinel-2 mission (S2A / S2B) offers the best revisit time and the best spatial resolution for most spectral bands. Landsat 8 data is used to complement Sentinel-2 data.

Also, this harmonisation process significantly improves the revisit time: the theoretical number of acquisitions of this virtual constellation (95 products / year) is increased by 30 % with respect to Sentinel-2 (S2A & S2B) only acquisitions (73 products / year).

However, an additional constraint has been set on the time lineless of this data fusion process, requiring that the product of a 10.0 m Landsat 8 image solely relies on the observations made in the past. In this context, the OLI synthetic surface reflectance measurements at MSI spatial resolution are result of time-based statistical prediction at pixel level. The main shortcomings are twofold,

- Correctness of the cloud shadow classification maps associated with inputs images involved in the prediction algorithms is expected
- Past data are in some cases not appropriate to fully characterize the biophysical processes (phenological stages) or even predict abrupt changes.

It is worth noting that for situation discussed in the second point, a dedicated quality assurance information has been added into the product to flag inconsistent measurements.

A sensor is sensing different characteristics of a landscape, basically break into large scale and small-scale features.

From image processing point of view, the large-scale features are regular, not necessarily uniform, with no discontinuity whilst the small-scale features are associated to contour and texture. The large / small scale features are attributed to respectively low and high spatial frequency content.

The Sen2Like approach relies on this basic decomposition for improving the spatial resolution of Landsat 8 OLI data at a given date.

The Landsat 8 existing large-scale features are complemented with predicted Sentinel-2 small scale features, at least small-scale features not captured by Landsat 8, as described with the following equation.

$$L8^{10m} = S_{L8}^{30\to10} + D_{S2-L8}^{10m} \text{ (Equation 1)}$$

Where:

- $L8^{10m}$ is the final Landsat 8 image at the Sentinel-2 spatial resolution, deconvolution from 30.0 m to 10.0 m
- $S_{L8}^{30\to10}$ is the original Landsat 8 image resampled from 30.0 m to 10.0 m by using bilinear interpolation, it is associated to the phase of Signal,
- D_{S2-L8}^{10m} is the image of differences, differences between 30.0 m and 10.0 m spatial resolution, this information is predicted by using Sentinel-2 data, it is associated to the amplitude of signal,

For a date t, a measurement in $S_{L8}^{30\to10}$ at image coordinates (i_0, j_0) can be expressed as an image convolution operation between the $L8^{10m}$ image and a low pass filter FS. For a small 3 x 3 window w, following mathematical relationship comes up:

$$L8_{t,i_0,j_0}^{30\to10} = L8_{t,w}^{10m} * FS = \frac{1}{9}\sum_{S_{i,j}\in W} S_{i,j} = \frac{1}{9}\left(\sum_{\substack{S_{i,j}\in W \\ (i,j)!=(i_0,j_0)}} S_{i,j} + S_{i_0,j_0}\right) = R + \frac{1}{9}S_{i_0,j_0} \text{ (Equation 2)}$$

Where main quantities have been defined before. If any, it is worth noting that $S_{i,j}$ are unknown.

Similarly, for a date t, the resolution difference between the two 10.0 and 30.0 m images, corresponding to details seen by S2 and not seen by LS8, D, can be appreciated as an image convolution operation between the $L8^{10m}$ image and a high pass filter FD.

Also, for a small 3 x 3 window w, still centred at image coordinates (i_0, j_0), the following mathematical relationship comes up:

$$D_{t,i_0,j_0}^{10m} = L8_w^{10m} * FD$$

Where $L8_w^{10m}$ is the measurement window from the final Landsat 8 image. The resolution difference, D_{t,i_0,j_0}^{10m}, is unknown.

The assumption is made that this quantity can be estimated by using Sentinel-2 observations performed in the past (before the date t). In this context, the following mathematical relationship can be proposed:

$$D_{t,i_0,j_0}^{10m} = f_l((S2_D)_{t-1}, \dots, (S2_D)_{t-K},)_{i_0,j_0} + \varepsilon_{i_0,j_0} \text{ (Equation 3)}$$

Where:

- $(S2_D)_{t-i} = (S2_{t-i,w}^{10m} - S2_{t-i,w}^{10m} * FS)$ with $i \in [1, K]$
- ε_{i_0,j_0} is the error term
- f_l is the best linear model prediction function minimizing error term

Also, with reference to (Eq.1, Eq.2), the resolution difference image for the concerned scaling, can be expressed as follow:

$$D^{10m}_{t,i_0,j_0} = S_{i_0,j_0} - \frac{1}{9}\left(\sum_{\substack{S_{i,j}\in W \\ (i,j)!=(i_0,j_0)}} (S_{i_0,j_0} + S_{i,j})\right) = \frac{8}{9}S_{i_0,j_0} - R$$

Following linearity assumption, one can adopt for simplicity,

$$S_{i_0,j_0} = \hat{S}_{i_0,j_0} + \varepsilon_{i_0,j_0}$$

It is now possible to compute the error term of the process, $\left|D^{10m}_{t,i_0,j_0} - \widehat{D^{10m}_{t,i_0,j_0}}\right|$ as follow:

$$Error_{term} = \left|(\frac{8}{9}\,\varepsilon_{i_0,j_0} - \frac{1}{9}\sum_{\substack{S_{i,j}\in W \\ (i,j)!=(i_0,j_0)}} (\epsilon_{i,j}))\right|$$

The following images have been extracted from the MGRS 31TFJ Sen2Like dataset. A side-by-side comparison of surface reflectance images for different landscapes (Valley, Field, Salt) shows the value added of the Sen2Like processing. Compared to 30.0 m data, the 10.0 m process enhances the image contours and enriched the image texture. Furthermore, the noise, a major drawback of this kind of approach is very limited.

Valley

Figure 3: Landsat 8, 10.0 m / 30.0 m side by side comparison, Valley type regions.

Figure 4: Landsat 8, 10.0 m / 30.0 m side by side comparison, Crop fields type regions.

Figure 5: Landsat 8, 10.0 m / 30.0 m side by side comparison, Salt type regions.

2.3 Processing details

The software is composed by "Thematic" blocks, one for each main algorithmic step to be implemented, and by "Generic" building blocks dedicated to data access, packaging, etc.

Concerning the geometric processing, Table 3 lists, depending on the band group defined in

Table 2, the applicability of each sub-processing (Framing, Co-registration, QC check). The initial MGRS tile geographical definition is always kept for subsequent processing, and then no framing is applied on Sentinel-2 data. Within the same frame, the co-registration process shifts image data depending on geolocation errors. If errors are too strong, a situation where missing pixel in margin area might be theoretically observed. Experience has shown that this situation happens only rarely. Nevertheless, it is worthy to highlight that nominally Sentinel-2 tiles already include 10km of overlap located at East and South sides.

Table 3: Sen2Like geometric processing applicability.

Processing	Geometric Processing + Geometric Check					
	Framing		Co-registration		QC Check *	
Mission	LS8	S2	LS8	S2	LS8	S2
Coastal + SWIR	x		x	x	x	x
RGB	x		x	x	x	x
NIR	x		x	x	x	x
Pan + Thermal	x	N/A	x	N/A	x	N/A
Red Edge	N/A		N/A	x	N/A	x

* Geometric Check processing step after stitching if any.

As shown in table just hereafter, for some Sentinel-2, Landsat 8/9 specific bands, there is no processing applied. More inter-calibration, not listed in the table, is exclusively applied to Sentinel-2 data.

Table 4: Sen2Like NBAR, SBAF, Fusion processing applicability.

Processing	NBAR		SBAF		Fusion	
Mission	LS8	S2	LS8	S2	LS8	
Coastal + SWIR	(1)x	(1)x	x	x	(3)x	30 m > 20 m
RGB	x	x	x	x	x	30 m > 10 m
NIR	x	x	x	(2)x	x	30 m > 20 m
Pan + Thermal	-	-	-	-	-	
Red Edge	-	-	-	-	-	

(5) – Only SWIR, (2) – Only S2 B8A, (3) – Only SWIR

As discussed more in details in this document, the Sen2Like fusion process applied to one specific product observed at a given date (d) requires also a sample of products observed prior to the date (d). With a minimum number of 2 past products. Clearly the process is more reliable when these products are as close as possible from the date (d).

2.4 Operational modes

Beyond operational modes, the s/w optimizes the input data selection based on fundamental criteria defined in Ground Image Processing Parameter (GIPP) file. These GIPP parameters are notably:

- The Cloud coverage parameter to filter MGRS image tile contaminates with strong cloud coverage.
- The percentage coverage parameter to discard products with a very limited geographic overlap with respect to MGRS tile extent.
- The priority parameter to manage the Landsat collection tiers inventory structure (Real-Time (RT), Tiers 1, Tiers 2).

This mechanism has been developed to support single tile mode and is de facto used for multi tile mode.

The processing performed in each operational mode are defined in the software configuration and can be overridden by command line arguments (see 5.1.2.1 – Processing).

Operational modes only differ in the way inputs are provided.

2.4.1 Single-Tile Mode

The processor considers as input one MGRS tile and a time period (start time / end time). By using this mode, all corresponding LS8 (LS9) and S2 data are processed. An MGRS multi temporal stack (L2F / L2H) is delivered.

2.4.2 Multi-Tile Mode

The processor considers as input an Area Of Interest (AOI) contained in a GeoJSON file and a period of time (start time / end time). By using this mode, all MGRS S2 tiles and LS8 (LS9) scenes that overlap the AOI within a selected period are processed. An AOI based multi temporal stack (L2F / L2H) is delivered.

It is worth noting that there is no clipping of output data to match exactly the geographical coverage of input AOI.

2.4.3 Product Mode

The product mode allows to feed the processor directly with an input product, LS8 (LS9) or S2, and is mainly useful for debugging purpose, or for environments that provide already the single/multi tile mechanism. The production of a time series on a specific MGRS tile, like the single tile does, would then require to manually execute a run of the processor for each input product of the time series, respecting the order of the acquisition dates (as many runs as input products).

2.4.4 ROI Based Mode

This mode takes as input:

- a ROI as GeoJSON file that must be fully contained within a MGRS tile
- a period (start time / end time).

Only one MGRS tile can be processed and can be specified as parameter.

By using this mode, all S2 tiles of the MGRS tile and LS8 (LS9) scenes that fully contains the AOI within the selected period are processed. An MGRS multi temporal stack (L2F / L2H) is delivered. **Pixels located outside the ROI extent are considered as nodata pixels, as shown in** Figure 6**.**

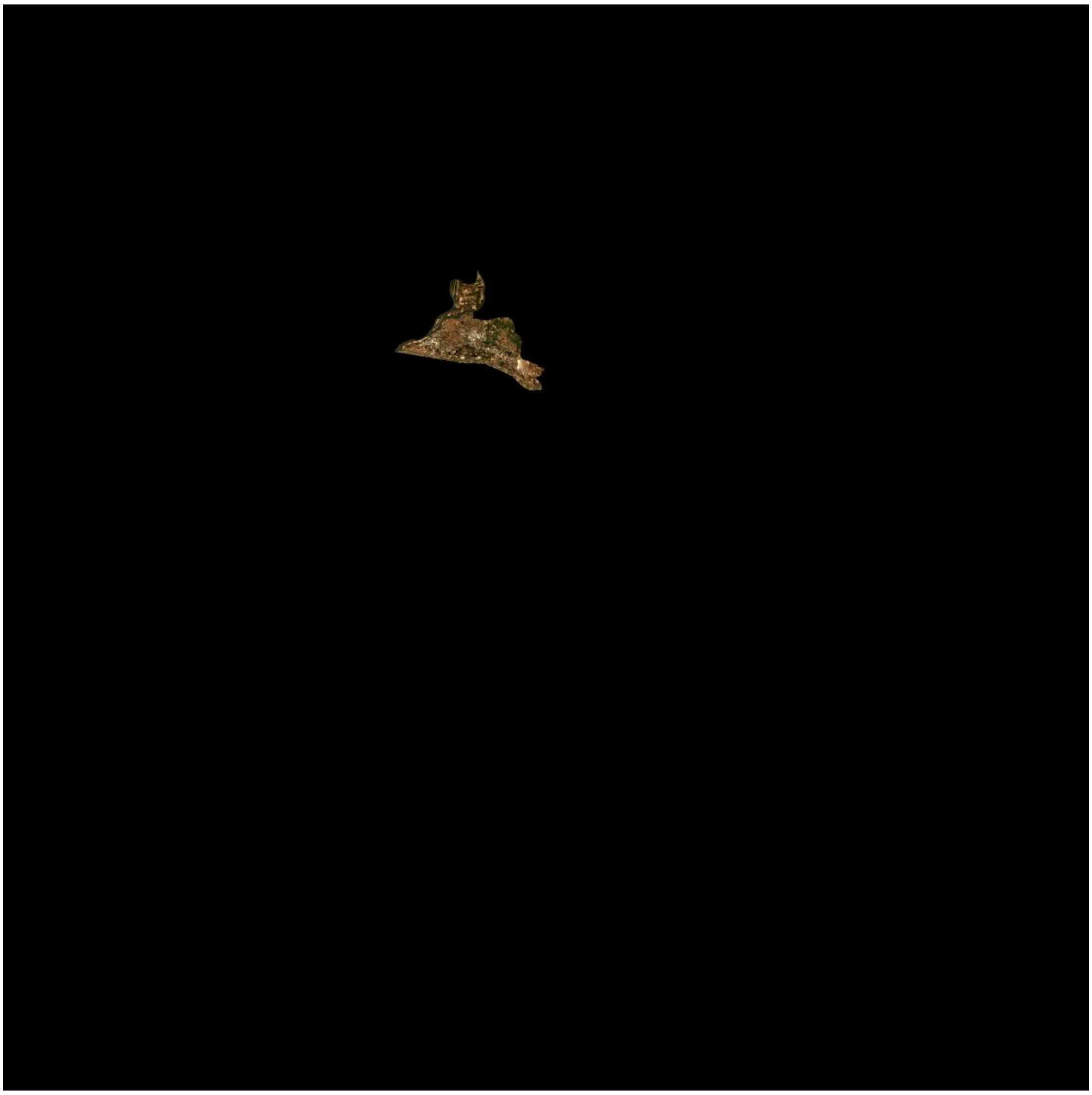

Figure 6: **Sen2Like ROI Based Mode product quicklook (T31TFJ).**

The reference of the ROI definition (GeoJSON) file is reported in the L2H(F)_QUALITY.xml file. The original GeoJSON file is also copied in the granule QI_DATA folder.

2.5 Design and Implementation

The software is developed in Python (version 3.12) as an open-source solution. The source code of Sen2Like is available in a Github repository, as part of a SNAP subfolder; https://github.com/senbox-org/Sen2Like

The software is composed of "Processing" blocks, one for each algorithm to be implemented, and of "Generic" building blocks dedicated to data access, packaging, etc. The "Processing" blocks present a generic interface. This simplifies the integration of new blocks, but also the switch of algorithms for a same thematic

An overview of the Sen2Like tool design is shown in the figure just here after. The idea is that each thematic block is implemented as a python class which presents a generic interface to the main program and uses internally the specific python packages dedicated to the thematic.

For example, the class "S2L_Atmcor" defines the thematic block for atmospheric correction, and is based on the "atmcor" package, which contains the Smac module and other atmospheric-oriented functions.

The data access layer and the orchestration of the software is supported by several classes and modules. For example, the configurations of the processing and the thematic blocks are managed in a specific class called "S2L_config".

The access to the product, its metadata and its data is managed through 4 different classes, "S2L_Product", "S2L_HLS_Product", "BaseReader" and "S2L_Image". The packaging of the output product is also implemented in tow class, called "S2L_PackagerL2F" and "S2L_PackagerL2H". The packaging of the output product metadata is implemented within "QI_MTD" module.

BaseReader and S2L_Product are generic classes that are specified to match different product types:

- BaseReader: Manage metadata access to a product type (Landsat8, Sentinel-2)
- S2L_Product; Represent a S2L product type in the software, with a generic interface (Landsat8, Sentinel-2)

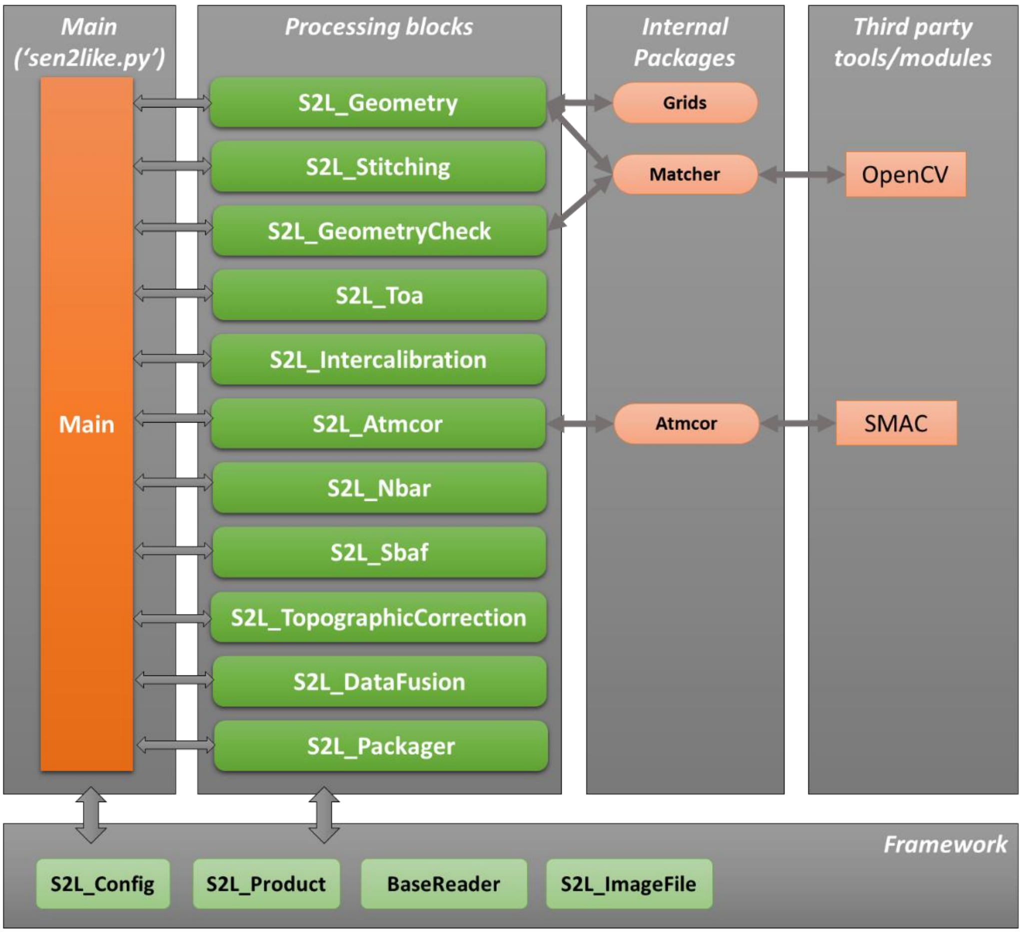

Figure 7: Overall software design

2.6 Limitations / disclaimers

The table lists the known limitations related to the usage of the Sen2Like software.

Table 5 – Limitations and disclaimers of the Sen2Like software.

Item	Description
Mixing Input data type: • Input S2 L2A & LS8 L1C (SMAC) • Input S2 L2A & S2 L1C (SMAC or Sen2Cor)	Accuracy lost due to this selection because for same thematic block different processing methods. Feedbacks from the S/W with a Warning is provided (TBC)
No clipping (multi-tile mode)	The AOI definition (GeoJSON) is not included into the L2H/L2F product. Sen2Like does not perform

Item	Description
	data clipping, i.e. sen2like does not set pixels outside AOI to nodata.
MGRS Selection using AOI (multi-tile mode) (Improvements Foreseen)	Even if Landsat WRS does not intersect the ROI the WRS corresponding to MGRS is selected
Data selection and cloud percentage	In multi tile mode and during data selection process, a product can be discarded because of high cloud cover score. A limitation is that the cloud cover can be related to an area in the tile that is finally outside from the AOI.
Missing data (No correction foreseen)	The co-registration process shifts the image data depending on geolocation errors. If errors are too strong, a situation where missing pixel in margin area might be theoretically observed.
Data Fusion Algorithm	The efficiency of the data fusion algorithm strongly depends on the reliability of the quality assurance mask. For instance, even if post processing is performed the quality of the L1C cloud mask remains degraded compared to the L2A Scene Classification mask. It results in inconsistencies in the predictive scheme, as discussed above in $2.2

3. INSTALLATION

3.1 Package installation

Installation of Anaconda

Please refer to the conda documentation: https://conda.io/projects/conda/en/stable/user-guide/install/linux.html#

Installation of Sen2Like code

- Using git:

```
git clone https://github.com/senbox-org/sen2like
cd sen2Like/sen2like
```

- Or from a downloaded archive from https://github.com/senbox-org/sen2like/releases:

```
unzip sen2like-x.y.z.zip
cd sen2like-x.y.z/sen2like
```

Create a conda virtual environment with required packages

```
conda env create --file environment.yml
```

you can also specify the name of the environment with:

```
conda env create --file environment.yml -n env_name
```

with `env_name` the name of the environment

Activate conda virtual environment

```
conda activate sen2like
```

Installation of OS dependencies

```
sudo apt-get install mesa-libGL
```

4. RUNNING TIME

The software contains a main python file, "sen2like.py", which is expected to be run through a command line with arguments and options.

```
usage: sen2like.py [-h] {product-mode,single-tile-mode,multi-tile-mode,roi-based-mode} ...

positional arguments:
  {product-mode,single-tile-mode,multi-tile-mode,roi-based-mode}
                        Operational mode
    product-mode        Process a single product
    single-tile-mode    Process all products on a MGRS tile
    multi-tile-mode     Process all products on a ROI
    roi-based-mode      Process all products that fully contains an ROI. The ROI
footprint must be FULLY INSIDE a MGRS tile.

options:
  -h, --help            show this help message and exit
```

The main argument is the operational mode to be used:

Main Argument	Description
single-tile-mode	Run the tool on a MGRS tile. Corresponding products will be loaded.
multi-tile-mode	Run the tool on a AOI defined in a GeoJSON. Corresponding MGRS tile will be inferred, and products will be loaded. It is equivalent to run a single-tile mode for each matching tile. In multi-tile mode, multiprocessing can be used to speed-up computation time
product-mode	Run the tool on a single product
roi-based-mode	Process all products that contains an ROI defined in a GeoJSON file within a period for only one MGRS tile. The ROI footprint must be fully contained inside an MGRS tile.

Depending on the choice of the operational mode, some options offered by the software can be different (see "Specific Options"). Other options are generic (see "Generic Options").

4.1 Specific Options

4.1.1 Single Tile Mode

Argument	Description

tile	Id of the MGRS tile to process
Options	**Description**
--start-date	Beginning of period (format YYYY-MM-DD)
--end-date	End of period (format YYYY-MM-DD)
--l2a	Indicates if Level-2A products have to be considered. If not set, Level-1C products will be processed (default: False).

4.1.2 Multi Tile Mode

Argument	Description
roi	GeoJSON file containing the AOI to process
Options	**Description**
--start-date	Beginning of period (format YYYY-MM-DD)
--end-date	End of period (format YYYY-MM-DD)
--jobs	Number of tiles to process in parallel
--l2a	Indicates if Level-2A products have to be considered. If not set, Level-1C products will be processed (default: False).

4.1.3 Product Mode

Argument	Description
product	Landsat8 L1 product path / or Sentinel2 L1C / L2A product path
tile	The tile on which is located the provided product.

4.1.4 ROI Based Mode

Argument	Description
roi	Path to JSON file containing the ROI to process
Options	**Description**

--start-date	Beginning of period (format YYYY-MM-DD)
--end-date	End of period (format YYYY-MM-DD)
--l2a	Indicates if Level-2A products have to be considered. If not set, Level-1C products will be processed (default: False).
--tile	tile to select if the ROI is contained by multiple MGRS tiles.

Warning: When using the ROI based mode, it is highly recommended to override archive_dir (5.1.2.2) configuration parameter (defined in config.ini/xml file) by using the generic option confParams with a directory name including the string "ROI". It prevents to overwrite already existing full image products stored in the existing archive_dir.

An alternative option consists in using a config file only dedicated to ROI based mode processing.

4.2 Generic Options

Generic Options	Description
--version	Display software version.
--conf	Sen2Like configuration file (default: SEN2LIKE_DIR/conf/config.ini) See chapter 4.4 for details.
--confParams	Overload parameter values (default: None). Given as a "key=value" comma-separated list. Example: --confParams "doNbar=False,doSbaf=False" *Parameters set in the confParams command line "option" supersede the parameters in the configuration file.*
--wd	Working directory (default: /data/production/wd)
--bands	Bands to process as coma separated list (default: ALL)
--allow-other-srs	Specify that selected input products can have a SRS that differs from the current MGRS tile
--refImage	Reference image (use as geometric reference) See chapter 5.2.1 for details.

--no-run	Do not start process and only list products (default: False). *Usually interesting before starting a single-tile or multi-tile processing to evaluate and verify all the products that will be processed.*
--intermediate-products	Generate intermediate products (default: False) For each processing block, each band, the intermediate output image is written saved into a file in the working directory.
--parallelize-bands	Process band in parallel
--debug	Display debug messages (default: False)
--no-log-data	Do not store timestamp in output log (default: False). Mainly used for log files comparison.

Note 1)
As shown in the product breakdown above, the process output both the 30-m dataset and the 10-m dataset. To get only 30-m dataset, the "doPackagerL2F" process should be disabled.

Note 2)
There are two important points for using the fusion process:

- ***It is not required to launch Sen2Like two times for one LS8/S2 product**, the Sen2Like command with "doPackagerL2F" process set to True allows to generate a product including both 10.0 m & 30.0 m bands.*
- *Prior launching Sen2Like on LS8 product with the "doPackagerL2F" option, **it is important to launch Sen2Like on at least the two S2 past products** observed as close as possible the LS8 product observation date.*

Note 3)
About --allow-other-srs: it is applied to the selection of products to process. That means it is not applied for the consecutive scene to search for stitching. To achieve stitching of consecutive scenes with different SRS, refers to the config parameter "same_utm_only" under section "Stitching" (5.1.2.9).

4.3 Examples

4.3.1 Single Tile Mode

```
python sen2Like.py single-tile-mode 31TFJ --wd ~/wd --refImage
/data/HLS/31TFJ/L2F_31TFJ_20170103_S2A_R008/L2F_31TFJ_20170103_S2A_R008_B04_10m.TIF
```

```
python sen2Like.py single-tile-mode 31TFJ --wd ~/wd --refImage
/data/HLS/31TFJ/L2F_31TFJ_20170103_S2A_R008/L2F_31TFJ_20170103_S2A_R008_B04_10m.TIF
--conf config.ini --start-date 2017-01-01 --end-date 2017-12-01 --no-run --
confParams “archive_dir=/data/S2L,coverage=0.5”
```

4.3.2 Multi Tile Mode

```
python sen2Like.py multi-tile-mode roi.geojson --wd ~/wd --refImage
/data/HLS/31TFJ/L2F_31TFJ_20170103_S2A_R008/L2F_31TFJ_20170103_S2A_R008_B04_10m.TIF
```

4.3.3 Product Mode

```
python sen2Like.py product-mode /eodata/Sentinel-
2/MSI/L1C/2017/01/03/S2A_MSIL1C_20170103T104432_N0204_R008_T31TFJ_20170103T104428.S
AFE --wd ~/wd --tile 31TFJ --bands B04
```

4.3.4 ROI Based Mode

```
python sen2Like.py roi-based-mode /tmp/on-rome.geojson --tile 33TTG --start-date
2017-01-01 --end-date 2017-12-01 --confParams “archive_dir=/data/S2L_ROI”
```

4.4 DEM Downloader utility

DEM Downloader does not work for now since prism-dem-open.copernicus.eu has been decommissioned.

The DEM Downloader utility (dem_downloader.py) allows to download Copernicus DEM and to create DEM that fit a MGRS tile extent and its UTM projection.

Created DEM file name matches the file name of DEM tile that sen2like expect thanks to its DEMRepository component (see 5.1.2.11).

4.4.1 Usage

It is available in the folder `aux_data/dem/`

```
dem_downloader.py [-h] --server_url SERVER_URL [--debug] MGRS_TILE_CODE
DEM_DATASET_NAME DEM_LOCAL_URL

    Create DEM for a MGRS Tile.
    The generated DEM is TIF file in the MGRS extend and its projected in the MGRS
tile EPSG (UTM).


positional arguments:
  MGRS_TILE_CODE        MGRS Tile code for witch to generate DEM, example 31TFJ
  DEM_DATASET_NAME      DEM dataset name, example COP-DEM_GLO-90-DGED__2022_1
```

```
  DEM_LOCAL_URL          Base output folder for generated DEM, example
/data/AUX_DATA/
                        Generated files are stored as follow:

/data/AUX_DATA/{DEM_DATASET_NAME}/Copernicus_DSM_{resolution}m_{MGRS_TILE_CODE}.TIF


options:
  -h, --help            show this help message and exit
  --server_url SERVER_URL
                        DEM server base URL (default: https://prism-dem-
open.copernicus.eu/pd-desk-open-access/prismDownload/)
  --debug, -d           Enable Debug mode (default: False)
```

To use it, `sen2like` folder MUST be in the PYTHONPATH environment variable.

Example:

```
PYTHONPATH=sen2like/ aux_data/dem/dem_downloader.py 33TTG COP-DEM_GLO-90-
DGED__2022_1 /data/AUX_DATA
```

Check README.md for additional information.

5. CONFIGURATION

5.1 Processor Configuration

The configuration of the tool is done by command-line arguments and by a configuration file.

The default location of the configuration file is `S2N2LIKE_DIR/conf`.

An example is provided in Appendix A.

5.1.1 Configuration File Format

Two configuration file formats are supported:

- INI file (.ini)
- GIPP file (.xml)

5.1.2 Configuration File Sections

In the configuration file the parameters are grouped into several sections, dedicated to the configuration of the orchestration of the processing blocks (on/off), the configuration of the data archives (inputs, outputs, auxiliary), and the internal configuration of the processing blocks.

5.1.2.1 Processing

Enable or disable a processing block based on value (`True, False`):

Parameter Name	Description	Type	Range
doGeometry	Run the geometric correction processing using KLT	Boolean	(True, False)
doStitching	Run the stitching processing	Boolean	(True, False)
doGeometryCheck	Run the geometric Quality Control using KLT	Boolean	(True, False)
doToa	Run the TOA conversion	Boolean	(True, False)
doInterCalibration	Run the Inter Calibration correction	Boolean	(True, False)
doAtmcor	Run the Atmospheric correction	Boolean	(True, False)

doNbar	Run Nbar correction processing	Boolean	(True, False)
doSbaf	Run the Sbaf correction processing	Boolean	(True, False)
doTopographicCorrection	Run the Topographic correction processing. If sen2cor3 is used, this parameter and sen2cor_topographic_correction MUST be set to True to enable sen2cor3 topographic correction	Boolean	(True, False)
doFusion	Run the Fusion processing	Boolean	(True, False)
doPackager	Run the packaging processing	Boolean	(True, False)
doPackagerL2H	Run the packaging processing for HLS.	Boolean	(True, False)
doPackagerL2F	Run the packaging processing for Fusion.	Boolean	(True, False)

5.1.2.2 Directories

Indicates path for special directories:

Parameter Name	Description	Type	Range
archive_dir	Where to store resulting products	Path	-
cams_dir	Where are located CAMS monthly files	Path	-
cams_daily_dir	Where are located CAMS daily files	Path	-
cams_hourly_dir	Where are located CAMS hourly files	Path	-
cams_climatology_dir	Where are located CAMS climatology files	Path	-
scl_dir	Where are located SCL maps files	Path	-

5.1.2.3 InputProductArchive

Describes parameters for product acquisition.
By default, three methods are described:

- **local**: products are stored in local
- **creodias**: products are located using the Creodias API
- **mixed archive**: products are located locally and remotely

Other access method can be defined by defining custom attributes to use other API. To define path, custom attributes can be defined in the configuration file.

In addition, these parameters are defined in the tool and can be used in brackets {}:

- mission: Landsat8 or Sentinel2
- tile: MGRS tile
- path: WRS path
- row: WRS row

5.1.2.3.1 Global

Parameter Name	Description	Type	Range
coverage	Define the coverage of the product tile in the interval	Float	[0, 1] (0-100%)
cloud_cover	Maximum cloud cover (%)	Int	0 – 100

Example:

```
coverage = 0.5
cloud_cover = 11
```

5.1.2.3.2 Local

Parameter Name	Description	Type	Range
coverage	Define the coverage of the product tile in the interval	Float	[0, 1] (0-100%)
base_url	Specify where the products are stored	Path	-
url_parameters_pattern_Sentinel2	Describe storage path for Sentinel-2 products	Pattern	-

url_parameters_pattern_Landsat8	Describe storage path for Landsat 8 products	Pattern	-

Example: with the following configuration:

```
base_url = /data/PRODUCTS
url_parameters_pattern_Sentinel2 = {base_url}/{mission}/{tile}
url_parameters_pattern_Landsat8 = {base_url}/{mission}/{path}/{row}
```

For a Sentinel-2 product on tile 31TFJ, the software will resolve:

```
url_parameters_pattern_Sentinel2 = /data/PRODUCTS/Sentinel2/31TFJ
```

5.1.2.3.3 Creodias API

Parameter Name	Description	Type	Range
base_url	Base address of the API	URL	-
location_Landsat8	Expression specifying Landsat 8 filter	Pattern	-
location_Sentinel2	Expression specifying Sentinel-2 filter	Pattern	-
url_parameters_pattern	API request URL. Special parameters between brackets are replaced by defined attributes	Pattern	-
thumbnail_property	Path in result JSON where product path is stored	Property	-
cloud_cover_property	Path in result JSON where cloud cover is stored	Property	-
S2_processing_level	Level of processing for considered products. Managed by software but can be specified here	Property	

Example: with the following configuration:

```
base_url = https://finder.creodias.eu/resto/api/collections
location_Landsat8 = path={path}&row={row}
location_Sentinel2 =
processingLevel={s2_processing_level}&productIdentifier=%25{tile}%25
url_parameters_pattern =
{base_url}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5B0%2C{cl
oud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=startDate&
sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
thumbnail_property = properties/productIdentifier
cloud_cover_property = properties/cloudCover
gml_geometry_property = properties/gmlgeometry
```

5.1.2.3.4 Mixed Archive

In this configuration mode, you can define for a mission its local or remote archive. So, you can define:

- Local S2 / Remote LS8/9
- Remote S2 / Local LS8/9

Local S2 Archive:

Parameter Name	Description	Type	Range
base_url_s2	Specify where the products are stored	Path	-
url_parameters_pattern_Sentinel2	Describe storage path for Sentinel-2 products	Pattern	-

Local Landsat Archive:

Parameter Name	Description	Type	Range
base_url_landsat	Specify where the products are stored	Path	-
url_parameters_pattern_Landsat8	Describe storage path for Landsat 8 products	Pattern	-
url_parameters_pattern_Landsat9	Describe storage path for Landsat 9 products	Pattern	-

Remote S2 Archive:

Parameter Name	Description	Type	Range
base_url_s2	Base address of the API	URL	-
location_Sentinel2	Expression specifying Sentinel-2 filter	Pattern	-
url_parameters_pattern	API request URL. Special parameters between brackets are replaced by defined attributes	Pattern	-
thumbnail_property	Path in result JSON where product path is stored	Property	-
cloud_cover_property	Path in result JSON where cloud cover is stored	Property	-
S2_processing_level	Level of processing for considered products. Managed	Property	

	by software but can be specified here		

Remote Landsat 8/9:

Parameter Name	Description	Type	Range
base_url_landsat	Base address of the API	URL	-
location_Landsat8	Expression specifying Landsat 8 filter	Pattern	-
location_Landsat9	Expression specifying Landsat 9 filter	Pattern	-
url_parameters_pattern	API request URL. Special parameters between brackets are replaced by defined attributes	Pattern	-
thumbnail_property	Path in result JSON where product path is stored	Property	-
cloud_cover_property	Path in result JSON where cloud cover is stored	Property	-

Example: Local Landsat / remote S2

```
##################################################################
# Mixed archive sample: local landsat and remote S2
# local landsat part
base_url_landsat = /data/PRODUCTS
url_parameters_pattern_Landsat8 = {base_url_landsat}/{mission}/{path}/{row}
url_parameters_pattern_Landsat9 = {base_url_landsat}/{mission}/{path}/{row}

# remote S2 part
base_url_s2 = https://finder.creodias.eu/resto/api/collections
location_Sentinel2 =
processingLevel={s2_processing_level}&productIdentifier=%25{tile}%25
url_parameters_pattern =
{base_url_s2}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5B0%2C
{cloud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=startDa
te&sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
thumbnail_property = properties/productIdentifier
cloud_cover_property = properties/cloudCover
gml_geometry_property = properties/gmlgeometry
```

Example: local S2 / Remote Landsat

```
##################################################################
# Mixed archive sample 2: local S2 and remote landsat
# local S2 part
base_url_s2 = /data/PRODUCTS
url_parameters_pattern_Sentinel2 = {base_url_s2}/{mission}/{tile}

# remote landsat part
```

```
base_url_landsat = https://finder.creodias.eu/resto/api/collections
location_Landsat8 = path={path}&row={row}
location_Landsat9 = path={path}&row={row}
url_parameters_pattern =
{base_url_landsat}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5
B0%2C{cloud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=st
artDate&sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
thumbnail_property = properties/productIdentifier
cloud_cover_property = properties/cloudCover
gml_geometry_property = properties/gmlgeometry
```

5.1.2.4 Geometry

Define parameters for geometric correction.

Parameter Name	Description	Type	Range
reference_band	The reference band to use for geometric correction. Default value is B04.	String	B01 – B12
doMatchingCorrection	Apply the matching correction	Boolean	(True, False)
doAssessGeometry	list of bands for which assessment is done	String	list of bands separated by comma
references_map	The path to the JSON file that contains for each MGRS tile, the file path of the corresponding reference image. (See details in $5.2.1.) Can be set to None.	Filepath	-
force_geometric_correction	Force geometric correction even if the Sentinel-2 product is refined (with Sentinel-2 Global Reference Image). Default True	Boolean	(True, False)

5.1.2.5 Atmcor

Describe the atmospheric method to be used.

Parameter Name	Description	Type	Range
use_sen2cor	Use sen2cor to do atmospheric correction if set to True. Use SMAC otherwise.	Boolean	(True, False)

sen2cor_path	Path to the sen2cor process.py file.	Path	-

5.1.2.6 Nbar

Describe the Nbar method to be used.

Parameter Name	Description	Type	Range
nbar_methode	The method to get coefficient to build constant.	String	{ROY,VJB}
vjb_coeff_matrice_dir	The path to the VJB coefficient file directory (only used by VJB method).	Path	-

5.1.2.7 Sbaf

Describe the adaptative SBAF configuration parameters.

Parameter Name	Description	Type	Range
adaptative	Activate the adaptative SBAF method.	Boolean	(True, False)
adaptative_band_candidates	list of bands for which adaptative method is applied	String	List of S2 band names separated by comma

5.1.2.8 Fusion

Define parameters for fusion processing.

To control fusion quality, a validity threshold mask is produced. Values of this mask are defined based on relative differences between l2fimage (returned at harmonised resolution) and l2himage: ((l2fimage – l2himage) / l2himage). If relative difference value is below a threshold value ('fusion_auto_check_band'), the L2F image pixel is valid.

This process is applied on only one image band specified with 'fusion_auto_check_band' parameter.

Parameter Name	Description	Type	Range

predict_method	Predict method to use (predict or composite using most recent valid pixels)	String	(predict, composite)
predict_nb_products	Number of products needed by predict method If not enough products in *predict_nb_products* It is the nearest product in the past that is considered	Int	1 - N
fusion_auto_check_band	Band of the validity threshold mask.	String	B01-B12
fusion_auto_check_threshold	Maximum percentage threshold to be valid in the mask.	Float	[0, 1] (0-100%)

Note 3)
In the current implementation, predict_nb_product parameter value does not exceed 2.

5.1.2.9 Stitching

Define parameters for stitching processing.

Parameter Name	Description	Type	Range
same_utm_only	Specify if consecutive scenes to stitch MUST be in the same SRS/UTM zone. Default to True	Bool	

5.1.2.10 TopographicCorrection

Define parameters for topographic correction processing.

Parameter Name	Description	Type	Range
topographic_correction_limiter	Specify the maximum value of topographic correction factor to apply	Float	
apply_valid_pixel_mask	Filter pixel to valid pixel using valid pixel mask for which apply correction. Useful to not apply correction to cloudy pixels for example.	Bool	True, False
sen2cor_topographic_correction	Enable sen2cor3 topographic correction.	Bool	False

	Notice that both parameters, sen2cor_topographic_correction and doTopographicCorrection MUST be set to True to enable sen2cor3 topographic correction		

5.1.2.11 DEMRepository

Define parameters for DEMRepository component.
DEMRepository is responsible to access DEM for TopographicCorrection.

Parameter Name	Description	Type	Range
dem_folder	Base folder path that hosts DEM dataset	Path	
dem_dataset	DEM dataset name	String	
src_dem_resolution	Resolution of DEM to use in the DEM dataset	int	

5.1.2.12 OutputFormat

Define modifier for written image file.

Parameter Name	Description	Type	Range
gain	Gain multiplier for output image (default: 10000)	Int	1 - N
offset	Offset to add to the output image (default: 1000)	Int	0 - N
output_format	Format of output images	String	(COG, GTIFF, 'JPEG2000)

5.1.2.13 COGOptions

Parameter Name	Description	Type	Range
interleave	Interleave mode	String	-
internal_tiling	Internal tiling value	Int	-
internal_overviews	Internal overviews value	Int	-
downsampling_levels_RES	Downsampling levels for resolution RES	Int List	-

downsampling_levels	Type of downsampling levels	String	-
resampling_algo_MASK	Resampling algorithm for mask	String	-
resampling_algo	Resampling algorithm	String	-
compression	Compression method	String	-
predictor	Predicator value	Int	-

5.1.2.14 JPEG2000Options

Parameter Name	Description	Type	Range
lossless_jpeg2000	Indicates if output products in JPEG2000 format are lossless.	Boolean	(True, False)

5.1.2.15 Multiprocessing

Define parameters for multiprocessing in multi-tile-mode.

Parameter Name	Description	Type	Range
number_of_process	Maximum number of processes to run in parallel	Int	1 - N

5.1.2.16 Packager

Define packaging parameters.

Parameter Name	Description	Type	Range
quicklook_jpeg_quality	Quality for outputs quicklooks in % (default: 95%)	Int	1 - 100
json_metadata	Indicates if metadata are also generated as JSON	Boolean	(True, False)

5.1.2.17 Runtime

This section is overridden during runtime and contains backup of computed values. Modifying this section will have no effect.

5.2 Auxiliary Data Configuration

5.2.1 HR BRDF data

As mentioned before, through configuration, it is possible to switch between two BRDF adjustment techniques (involved in the NBAR processing).

Whilst the ROY method coefficient set is hard coded (mission product classes), the per pixel HR BRDF coefficients are stored within a NetCDF file: one single file for a given tile and for a given 1-year period.

As an auxiliary data, the directory path pointing to the NetCDF file directory must be indicated in the configuration file by using *vjb_coeff_matrice_dir parameter (* HR BRDF directory path*)*

The file naming convention is defined according to Sentinel 2 Auxiliary Data format, as defined in the Sentinel-2 Product Specification Document (V14.9)[20].

As example, name of an HABA auxiliary file is as follow:

S2L_UV_AUX_HABA_100_20221027T114424_V20190105T103429_20191231T103339_T31TFJ_**M**LSS2_MO.nc

Where:

- 0100 : S/W Version (aux data generation)
- 20221216T00 : Generation date
- V20150622T000000 : Validity Start Date / Time
- 21000101T000000 : Validity End Date / Time
- T31TCJ : The MGRS TILE
- MLSS2 : Landsat and Sentinel-2 Missions contributes to the dataset, defined as indicated in table here below
- MOS : MODIS used a BRDF reference

More details regarding optional suffix are given in Table 6 below.

Table 6 - Auxiliary file optional suffix.

Optional Suffix	Description	Template*
Missions for HABA inversion : **_M**	Missions for HABA inversion **_M**xxxx xxxx = LS__, Landsat data is used xxxx = S2__, Sentinel-2 is used xxxx = LSS2, Both Landsat / Sentinel-2 is used.	_MLSS2

[20] https://sentinels.copernicus.eu/documents/247904/685211/S2-PDGS-TAS-DI-PSD-V14.9.pdf/3d3b6c9c-4334-dcc4-3aa7-f7c0deffbaf7?t=1643013091529

Missions for HABA BRDF Reference	Name of the mission _xx xx = VI (VIIRS) xx = MO (MODIS) xx = S3 (Sentinel 3)	_VI
Band Number in case of GeoTiff COG File, Netcdf image name	Band Number, S2L Convention _Bxx	_B10

The NetCDF file content is:

- **Coordinates**: geographical x, y coordinate
- **Data variables**:
  - The V0 and V1 band dependant coefficient for each s2 bands with the following keys:
    - *V_slope_{band name (ex: B02}*
    - *V_intercept_{band name (ex: B02}*
    - *R_slope_{band name (ex: B02}*
    - *R_intercept_{band name (ex: B02}*
    - *R_R2_{band name (ex: B02}*
    - *V_R2_{band name (ex: B02}*
    - V_RelStd_*{band name (ex: B02}*
    - R_RelStd_*{band name (ex: B02}*
  - A per pixel number of valid images involved in the BRDF inversion parameters
  - Ä per pixel minimum value of NDVI (over inversion temporal period)
  - Ä per pixel maximum value of NDVI (over inversion temporal period)

- **Attributes**:
  - TILE
  - SPATIAL_RESOLUTION
  - BANDS: list of band name (ex: Red)
  - BANDS_NUMBER: list of band id (ex: B04)

There is a specific tool dedicated to generation of HR BRDF data. For any information contact sen2like support.

5.2.2 Geometrical Reference Data

Sen2Like provides the capability to co-register the input products, including Sentinel-2 products. To enable this functionality, a reference image must be provided. The reference image shall be specified in the *references_map* file, or given within the Sen2Like command line option "--*refImage*". Note that if both are provided, the option "--*refImage*" has the priority.

The path of the *references_map* file must be specified in the configuration file (see parameter description in §0). The format of the *references_map* file is JSON and gives for each MGRS tile identifier (e.g. "31TFJ"), the path to the reference image file to be used. An example of this file is given in Appendix B.

This reference image shall be in the geometry of the MGRS tile to process (same extent). The resolution of the reference image is not fixed, but it is recommended to provide a resolution equal to the highest resolution of the input products, i.e. 10m.

Then Sen2Like will automatically resample the reference image when necessary, for instance for Landsat8 30m bands. The resampling process is done once, and then the resulting image, saved in the reference image directory, is directly reused.

The format of the reference image is usually JPEG2000 or GeoTIFF. But it can be any format compatible with the GDAL library.

At end, the following configuration must be verified:

- *reference_band*: the equivalent Sentinel-2 band to use for matching with the provided reference image (B04 by default)
- *doMatchingCorrection*: enable the co-registration correction
- *doAssessGeometry*: enable co-registration assessment
- *references_map:* path to the references map file

For more details on these parameters, see section 0.

5.2.3 SCL map

When processing Sentinel-2 L1C products, the corresponding Scene Classification (SCL) map available in Sentinel-2 L2A products can be used to generate the valid pixels mask associated with different Sen2like processing blocks (e.g. fusion).

Please note that this is a temporary solution for the processing of Sentinel-2 L1C product without using Sen2Cor 3.0 for atmospheric correction.

This database of L2A SCL maps needs to be populated by the user, with SCL maps in Cloud Optimized GeoTIFF format (.tif extension).

The access to this database is then configured with the *scl_dir* parameter (see 5.1.2.2).

If *scl_dir* is not set, the valid pixels mask is computed based on Sentinel-2 L1 cloud mask.

Characteristics of the SCL database are given hereafter.

SCL database

Description:

One directory per tile (e.g. 31TFJ).

Each directory contains SCL COG files generated with Sen2Cor 2.10 at 60 m resolution with the "sc_cog" option, following the naming convention: T31TFJ_20201228T104349_SCL_60m.tif

Examples of directories:

31TFJ 12SVB

Examples of SCL COG file:
31TFJ/T31TFJ_20201228T104349_SCL_60m.tif

5.2.4 CAMS data

For enabling the use of CAMS data in Sen2Like, the database must be prepared.

> ***Note***
>
> *The CAMS monthly database for the Year 2020 is available here:*
>
> *http://185.178.85.51/CAMS/*

When ready, the access to this database is configured with the *cams_dir* parameter (see 5.1.2.2).

It CAMS is not configured, a default AOT value will be used.

Sen2Like is able to retrieve atmospheric parameters from 4 type of databases, derived from ECMWF/CAMS:

- CAMS Monthly database (analysis)
- {CAMS Daily database (near real time)}
- {CAMS Hourly database (near real time)}
- {CAMS climatology database (climatology)}

Specificities of each database are given below.

CAMS Monthly database (analysis)

Description:

One directory per month, with a naming as 201704 for April 2017. Each directory contain a single NetCDF file with the following naming convention: CAMS_archive_aod550_tcwv_msl_gtco3_[reanalysis|analysis]_0H_6H_12H_18H_YYYY-MM.nc.

Each NetCDF file covers the whole month, with data every 6H, and contain data for the 4 parameters

Format of NetCDF file is such as provided by the ECMWF/CAMS API.

Examples of directories:

201701 201703 201705 201707 201709 201711 201801 201803 201805 201807 201809 201811 201901 201903 201905 201907 201909 201911 202001

Examples of NetCDF files:

201601/CAMS_archive_aod550_tcwv_msl_gtco3_reanalysis_0H_6H_12H_18H_2016-01.nc
201805/CAMS_archive_aod550_tcwv_msl_gtco3_analysis_0H_6H_12H_18H_2018-05.nc
201806/CAMS_archive_aod550_tcwv_msl_gtco3_analysis_0H_6H_12H_18H_2018-06.nc

{CAMS Daily database (analysis)}

Description:

One directory per day, with a naming as 20210417 for 17th of April 2021. Each directory contains a single NetCDF file with the following naming convention: CAMS_archive_aod550_tcwv_msl_gtco3_[analysis]_0H_6H_12H_18H_YYYY-MM-DD.nc
Each NetCDF file covers 4 times of the day (0h, 6h, 12h, 18h) and contain data for the 4 parameters

Format of this NetCDF file is such as provided by the ECMWF/CAMS FTP Near-Real-Time server.

Examples of directories:

20210201 20210202 20210203 20210204 20210205 20210206 20210207 20210208

Examples of NetCDF files:

20210501/CAMS_archive_aod550_tcwv_msl_gtco3_analysis_0H_6H_12H_18H_2021-05-01.nc
20210502/CAMS_archive_aod550_tcwv_msl_gtco3_analysis_0H_6H_12H_18H_2021-05-02.nc

CAMS Hourly database (near real time)

Description:

One directory per 12hours, with a naming as 2020040812 for 2020/04/08 12:00. Each directory contains a list of NetCDF file, one per parameter and per hour, with a forecast until 12hours.

Format of this NetCDF file is such as provided by the ECMWF/CAMS FTP Near-Real-Time server.

Examples of directories:

2020040700 2020040712 2020040800 2020040812 2020040900 2020040912 2020041000 2020050300

Examples of files (for 1 directory):

2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_001_aod550.nc

2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_001_gtco3.nc

2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_002_aod550.nc

2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_002_gtco3.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_003_aod550.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_003_gtco3.nc
…
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_010_aod550.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_010_gtco3.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_011_aod550.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_011_gtco3.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_012_aod550.nc
2020040912/z_cams_c_ecmf_20200409120000_prod_fc_sfc_012_gtco3.nc

CAMS Climatology database (climatology)

Description:

Climatology files are generated manually from CAMS archive. For each parameter, they present one value per day of year. The role of the climatology database is about having a backup strategy, i.e. if any valid data cannot be found in other databases.

Data file format:

The format is a TIF internal format. Size and extent of the data corresponds to official NetCDF data. Files are single date.

Example of files:

CAMS_Climatology_2010-2019_msl_DOY_355.tif
CAMS_Climatology_2010-2019_tcwv_DOY_001.tif

5.2.5 DEM

DEM storage tree MUST follow the following structure:

```
{dem_folder}/{dem_dataset}/Copernicus_DSM_{src_dem_resolution}m_{tile_code}.TIF
```

Where `tile_code` is a MGRS tile code.

Other path variables correspond to DEMRepository (5.1.2.11) config parameters.

Exemple:

```
/data/DEM
├── COP-DEM_GLO-90-DGED__2022_1
│   ├── Copernicus_DSM_90m_12SYH.TIF
│   ├── Copernicus_DSM_90m_31TFJ.TIF
│   ├── Copernicus_DSM_90m_33TTG.TIF
```

5.3 Processing Blocks Configuration

This chapter gives information that might be useful for integrating new processing blocks.

The orchestration of the chain of blocks is at the level of the main program ("Sen2Like.py" file). It starts with the ingestion of the input product, triggering metadata and product information extraction. When the thematic blocks are eligible to a "**band by band" process**, the orchestrator can loop the processing to cover all the requested band, the band set being specified and by default it is all the band in the product that are processed.

The 'band by band' approach has been developed to ease parallelization in the deployment. The "band set" approach has been set up to ensure quick processing of expected band.

The design has considered two types of configurations:

- An internal configuration to allow the specification of a module in term of parameter and applicability
- An external configuration to allow orchestration of the different modules.

The list of the thematic blocks to be enabled for the processing is dynamically set up from the external configuration (ON/OFF switches) and the internal configuration.

The internal configuration lists the names of the available classes that implement thematic blocks. In addition, they are some parameters, like the applicability for LS8/S2.

In the external configuration, for each thematic block, the user can add an ON/OFF switch. The name of the parameter is the name of the class, but where the prefix 'S2L_' is replaced by 'do'. For example, the ON/OFF parameter of the atmospheric correction block, implemented through the 'S2L_Atmcor' class, is: 'doAtmcor'.

Table 7: Example of internal configuration (declaration of building blocks)

```
PROC_BLOCKS['S2L_Geometry'] = {'extension': '_REFRAMED.TIF', 'applicability':
'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Stitching'] = {'extension': '_STITCHED.TIF', 'applicability':
'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_GeometryCheck'] = {'extension': None, 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Toa'] = {'extension': '_TOA.TIF', 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_InterCalibration'] = {'extension': '_INTERCAL.TIF', 'applicability':
'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Atmcor'] = {'extension': '_SURF.TIF', 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Nbar'] = {'extension': '_BRDF.TIF', 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Sbaf'] = {'extension': '_SBAF.TIF', 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_TopographicCorrection'] = {'extension': '_TOPOCORR.TIF', 'applicability':
'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_PackagerL2H'] = {'extension': None, 'applicability': 'L8_L9_S2_Prisma'}
PROC_BLOCKS['S2L_Fusion'] = {'extension': '_FUSION.TIF', 'applicability': 'L8_L9_Prisma'}
PROC_BLOCKS['S2L_PackagerL2F'] = {'extension': None, 'applicability': 'L8_L9_S2_Prisma'}
```

APPENDIX A: EXAMPLE OF CONFIGURATION FILE

```
[Processing]
doGeometry = True
doStitching = True
doGeometryCheck = True
doToa = True
# doInterCalibration must be set to True to generate consistent S2A-S2B timeseries
doInterCalibration = True
doAtmcor = True
doNbar = True
doSbaf = True
doTopographicCorrection = True
doFusion = True
doPackagerL2H = True
doPackagerL2F = True

[Directories]
archive_dir = /data/S2L

cams_dir = /data/CAMS/monthly
cams_daily_dir = /data/CAMS/daily
cams_hourly_dir = /data/CAMS/hourly
cams_climatology_dir = /data/CAMS/climatology/v1

scl_dir = /data/AUX_DATA/SCL_maps_2.10

[InputProductArchive]
# global InputProductArchive params
coverage = 0.5
cloud_cover = 11

##################################################################
# Local only
base_url = /data/PRODUCTS
url_parameters_pattern_Sentinel2 = {base_url}/{mission}/{tile}
url_parameters_pattern_Landsat8 = {base_url}/{mission}/{path}/{row}
url_parameters_pattern_Landsat9 = {base_url}/{mission}/{path}/{row}

##################################################################
# Creodias only
; base_url = https://finder.creodias.eu/resto/api/collections
; location_Landsat8 = path={path}&row={row}
; location_Landsat9 = path={path}&row={row}
; location_Sentinel2 = productIdentifier=%25T{tile}%25
; url_parameters_pattern =
{base_url}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5B0%2C{cl
oud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=startDate&
sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
; thumbnail_property = properties/productIdentifier
; cloud_cover_property = properties/cloudCover
; gml_geometry_property = properties/gmlgeometry

##################################################################
# Mixed archive sample: local landsat and remote S2
# local landsat part
;base_url_landsat = /data/PRODUCTS
;url_parameters_pattern_Landsat8 = {base_url_landsat}/{mission}/{path}/{row}
;url_parameters_pattern_Landsat9 = {base_url_landsat}/{mission}/{path}/{row}
```

```
# remote S2 part
;base_url_s2 = https://finder.creodias.eu/resto/api/collections
;location_Sentinel2 = productIdentifier=%25T{tile}%25
;url_parameters_pattern =
{base_url_s2}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5B0%2C
{cloud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=startDa
te&sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
;thumbnail_property = properties/productIdentifier
;cloud_cover_property = properties/cloudCover
;gml_geometry_property = properties/gmlgeometry

##################################################################
# Mixed archive sample 2: local S2 and remote landsat
# local S2 part
;base_url_s2 = /data/PRODUCTS
;url_parameters_pattern_Sentinel2 = {base_url_s2}/{mission}/{tile}

# remote landsat part
;base_url_landsat = https://finder.creodias.eu/resto/api/collections
;location_Landsat8 = path={path}&row={row}
;location_Landsat9 = path={path}&row={row}
;url_parameters_pattern =
{base_url_landsat}/{mission}/search.json?maxRecords=1000&_pretty=true&cloudCover=%5
B0%2C{cloud_cover}%5D&startDate={start_date}&completionDate={end_date}&sortParam=st
artDate&sortOrder=ascending&status=all&{location}&dataset=ESA-DATASET
;thumbnail_property = properties/productIdentifier
;cloud_cover_property = properties/cloudCover
;gml_geometry_property = properties/gmlgeometry


[Geometry]
reference_band = B04
doMatchingCorrection = True
doAssessGeometry = B04
references_map = /data/References/references_map.json
force_geometric_correction = True

[Atmcor]
use_sen2cor = False
sen2cor_path = ../sen2cor/process.py

[Nbar]
nbar_methode = VJB
vjb_coeff_matrice_dir = /data/Belen

[Sbaf]
adaptative = False
adaptative_band_candidates = B04,B11,B12

[fusion]
# predict_method: predict or composite (most recent valid pixels)
predict_method = predict
predict_nb_products = 2
fusion_auto_check_band = B04
fusion_auto_check_threshold = 0.1

[Stitching]
same_utm_only = True
```

```
[TopographicCorrection]
topographic_correction_limiter = 4.0
apply_valid_pixel_mask = True

[DEMRepository]
# Expect to get DEM from
{dem_folder}/{dem_dataset}/Copernicus_DSM_{src_dem_resolution}m_{tile_code}.TIF
dem_folder = /data/AUX_DATA/
dem_dataset = COP-DEM_GLO-90-DGED__2022_1
src_dem_resolution = 90

[OutputFormat]
gain = 10000
offset = 1000
output_format = COG

[COGoptions]
interleave = PIXEL
internal_tiling = 1024
internal_overviews = 1024
downsampling_levels_10 = 2 6 12 36
downsampling_levels_15 = 2 4 8 24
downsampling_levels_20 = 3 6 18
downsampling_levels_30 = 2 4 12
downsampling_levels_60 = 2 6

downsampling_levels = variable
resampling_algo_MASK = MODE
resampling_algo = AVERAGE
compression = LZW
predictor = 1

[JPEG2000options]
lossless_jpeg2000 = True

[Multiprocessing]
number_of_process = 5

[Packager]
quicklook_jpeg_quality = 75
json_metadata = True
```

APPENDIX B: EXAMPLE OF REFERENCES MAP FILE

```
{
  "30SWJ":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000094_T30SWJ_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000094_T30SWJ_B04.jp2",
  "30TXQ": "/data/References/30TXQ/L2F_30TXQ_20190822_S2B_R094_B04_10m.TIF",
  "32TNS":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000065_T32TNS_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000065_T32TNS_B04.jp2",
  "32TMR":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000065_T32TMR_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000065_T32TMR_B04.jp2",
```

```
  "36MXE":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_SGS__20160217T115519_A003421_T36MXE_N02.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_SGS__20160217T115519_A003421_T36MXE_B04.jp2",
  "34RGS":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MTI__20180617T111214_A015591_T34RGS_N02.06/IMG_DATA/S2A_OPER_MSI_L1C_TL_MTI__20180617T111214_A015591_T34RGS_B04.jp2",
  "20MRB":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_SGS__20160806T192619_A005870_T20MRB_N02.04/IMG_DATA/S2A_OPER_MSI_L1C_TL_SGS__20160806T192619_A005870_T20MRB_B04.jp2",
  "12SVB":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_SGS__20160420T214215_A004328_T12SVB_N02.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_SGS__20160420T214215_A004328_T12SVB_B04.jp2",
  "32TQM":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000122_T32TQM_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000122_T32TQM_B04.jp2",
  "35WMQ":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000022_T35WMQ_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000022_T35WMQ_B04.jp2",
  "31TCJ":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000051_T31TCJ_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000051_T31TCJ_B04.jp2",
  "31TFJ":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000008_T31TFJ_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000008_T31TFJ_B04.jp2",
  "34TCR":
"/data/References/GRI/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000036_T34TCR_N01.01/IMG_DATA/S2A_OPER_MSI_L1C_TL_MPS__20161018T120000_A000036_T34TCR_B04.jp2"
}
```


APPENDIX C: PRISMA INTEGRATION

Introduction

This appendix describes how to use primsa4sen4like “pre-processor” to transform Prisma native L1 products into Sentinel-2/Prisma (S2P) L1C products compatible with sen2like. Sentinel-2 PRISMA (S2P) L1C products are spectrally aggregated into 13 Sentinel-2 bands and projected into L1C Sentinel-2 geometry (without refinement using Sentinel-2 GRI).

Please note that prisma4sen2like is a prototype and that the output products are internal intermediate products that are not yet mature, especially in terms of metadata and quality of geographic projection.

Conda environment:

A recent version of conda is required to prepare prisma4sen2like conda environment as it is based on Python 3.10 (unlike sen2like). For now, prisma4sen2like needs a distinct conda environment, slightly different from the conda environment of sen2like.

Sentinel-2 Prisma local archive:

S2P L1C products shall be processed using the local Sentinel-2 archive folder (see configuration in chapter 5.1.2.3.2) because Prisma mission does not have a dedicated archive folder. Nevertheless, Sentinel-2A/B and S2P L1C products can be mixed in the local Sentinel-2 archive folder.

Installation:
After having retrieved the code from sen2like github repository as described in the chapter 3.1, goes into the prisma4sen2like folder and creates its conda environment:

```
cd sen2like/prisma4sen2like
conda env create -f environment.yml
```

To run prisma4senlike, activate its conda environment:

```
conda activate prisma
```

Verify it works fine with help command:

```
PYTHONPATH=prisma python prisma/main.py -h
```

That should output the usage help

```
usage: main.py [-h] PRISMA_L1_FILE DESTINATION_FOLDER WORKING_DIR

positional arguments:
  PRISMA_L1_FILE      Prisma L1 he5 file
  DESTINATION_FOLDER  Generated S2P destination folder
  WORKING_DIR         Working directory

options:
  -h, --help          show this help message and exit
```

Usage

The program takes 3 arguments:
- PRISMA L1 product H5 file
- Product output folder
- Working directory.

Please note that the output and working directories must exist.

The main prisma module MUST be in the PYTHONPATH. To do that run the following command from prisma4senlike folder:

```
export $PWD/prisma
```

You can now run the program with

```
python prisma/main.py ARGS …
```

You can also do it in a single command:

```
PYTHONPATH=prisma python prisma/main.py ARGS …
```

Example:

```
PYTHONPATH=prisma python prisma/main.py
/data/Products/PRISMA/PRS_L1_STD_OFFL_20220714100507_20220714100511_0001.he5
/data/Products/S2P /data/wd
```

The program will output the path to the generated product.

Please note that prisma4sen2like can find automatically which MGRS tile (e.g. 33TVF) should be associated to the Prisma native product when creating the S2P L1C product.

To use the S2P L1C product generated in sen2like, copy or move the S2P L1C product in sen2like local archive for Sentinel-2 products in the proper tile directory (e.g. 33TVF), then run sen2like with the appropriate configuration to process it.