EOReader

an open source remote-sensing python library

Sertit

iCUBE

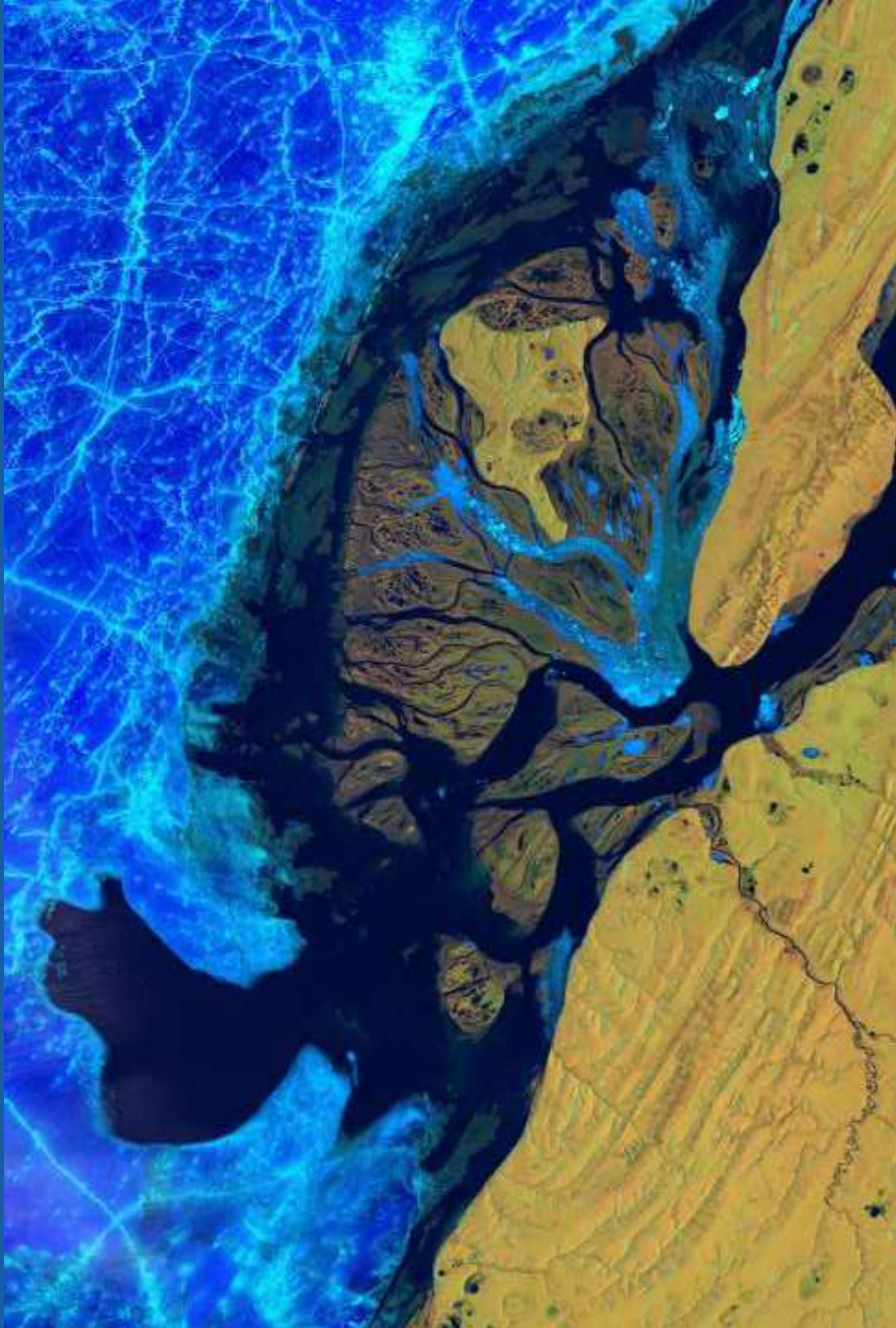
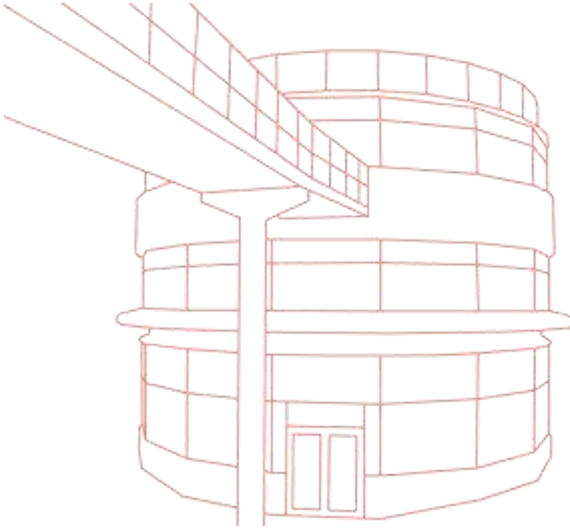


Table of Contents

- Context
- Main features
- About the project

Context

Context



ICube-SERTIT

- Technological and service platform of **ICube** Laboratory in **University of Strasbourg**
- More than **30** year experience of **valorisation** and **technological transfer** in space techniques and **Earth Observation** applications
- Production of **geo-information** for:

Environmental studies

Urban planning

Forest management, Natural ressource monitoring

Natural disaster and crisis management

24/7 Rapid Mapping Service

Context

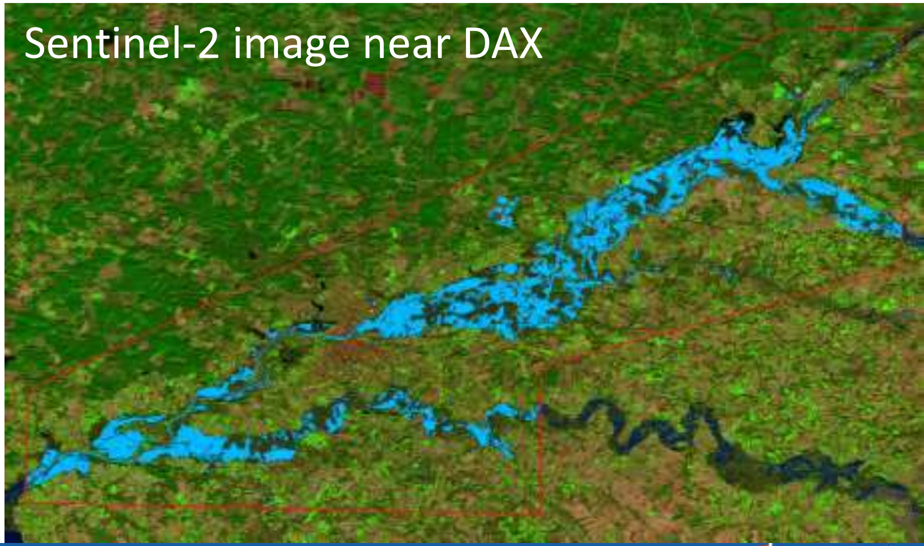
24/7/365
Rapid Mapping
Service

ICube-SERTIT

- **Rapid Mapping** Service

  a **risk and crisis** management service operational **24/7/365**

  - Uses **remote sensing images** (satellites and sometimes drones)
  - **Various** sensors: *~15 different, optical and radar*
  - **Various** delivery products: *flood, fire, landslides, etc.*
  - Delivery in **near-real-time**



Sentinel-2 image near DAX

Context

Motivation behind **EOReader**

- Satellite data: every sensor is different (bands, storage, …)
- Crucial to **harmonize** and **increase the reliability** of the production tools used in a **industrialized** framework (*ie. make them as **sensor-agnostic** as possible*):
  - the developer can focus on **core** tasks *(such as extraction)* without taking into account the sensor characteristics
  - New sensors are added **effortlessly** (if existing in EOReader) and **without any modification** of any tool
  - Maintenance and testing are **simplified** and the code is **more readable**

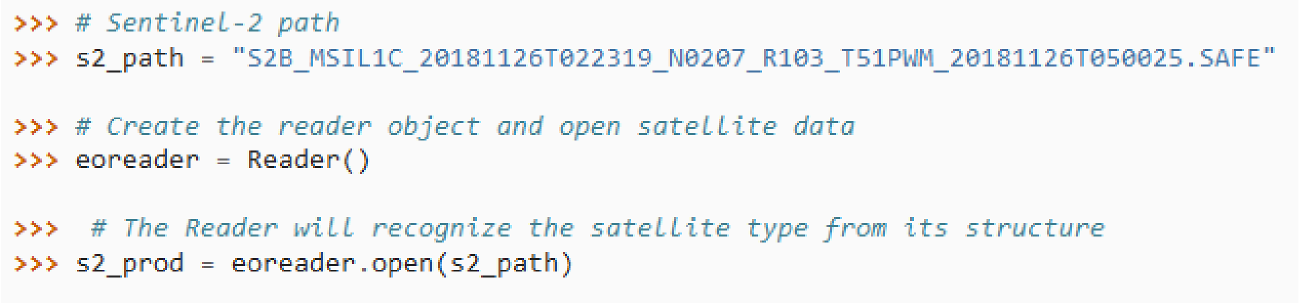
Main features

Available sensors

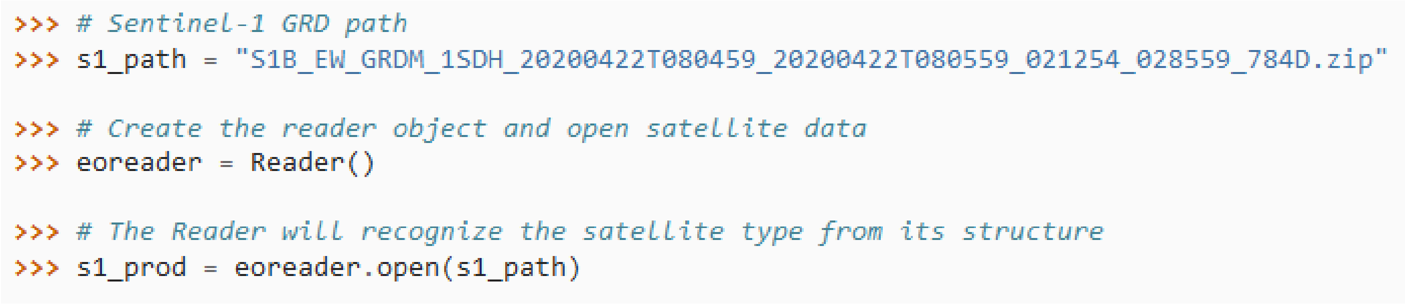
Optical sensors	SAR sensors
Sentinel-2 and **Sentinel**-2 Theia **Sentinel**-3 OLCI and SLSTR	**Sentinel**-1
Landsat 1 to 9 (MSS, TM, ETM and OLCI)	**COSMO-Skymed** 1st and 2nd Generation
PlanetScope	**TerraSAR-X**, **TanDEM-X** and **PAZ SAR**
Pleiades-Neo and **Pleiades** **SPOT** 6-7	**RADARSAT**-2 **RADARSAT**-Constellation
Vision-1	**ICEYE**
WorldView-2 to 4, **GeoEye**-1 *(and other Maxar sensors)*	**SAOCOM**-1

Open sensors

- EOReader opens the sensor products **agnostically**
  - Recognizes the sensor thanks to the product **name** and/or **structure** (by default)
- Example



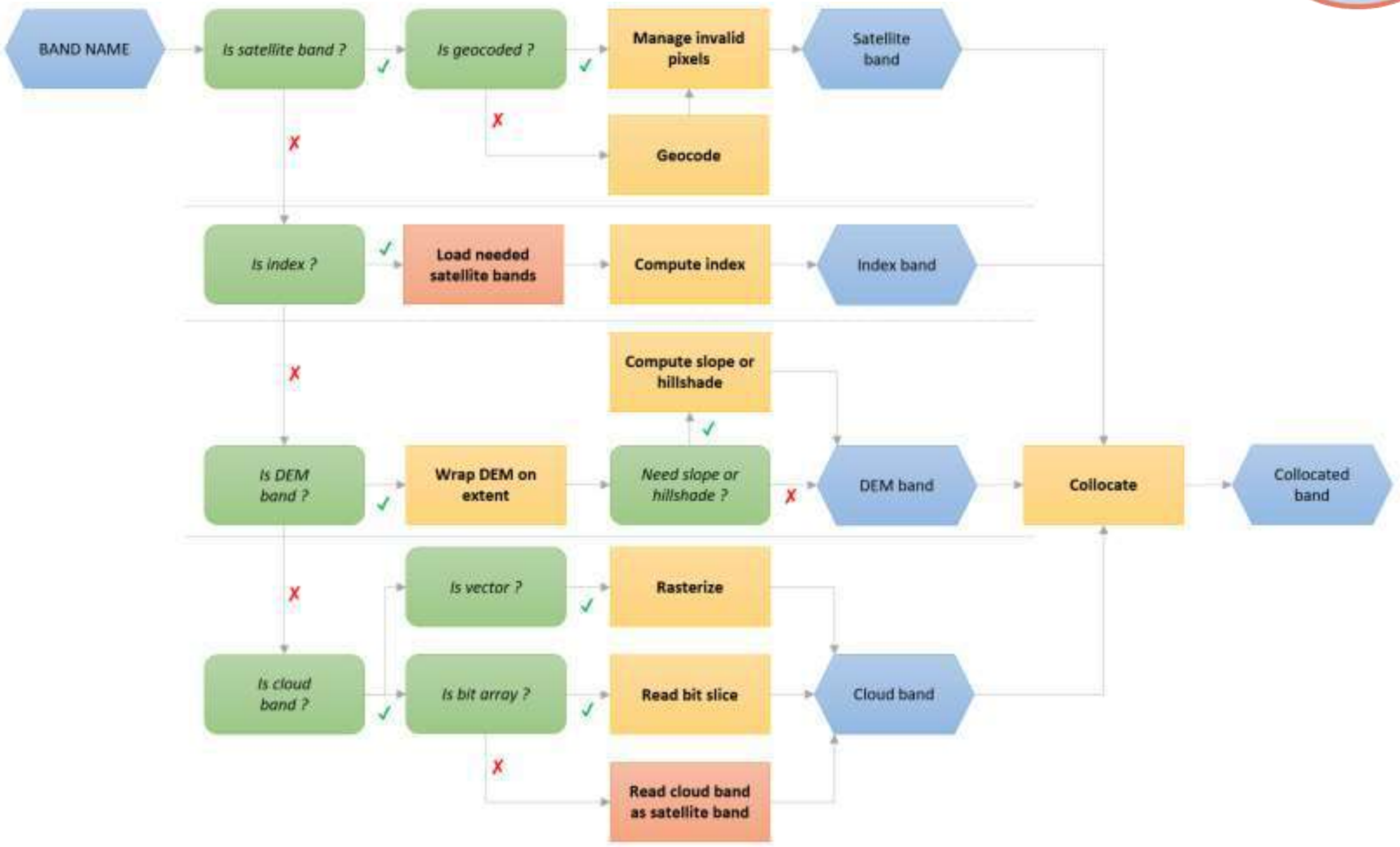
```
>>> # Sentinel-2 path
>>> s2_path = "S2B_MSIL1C_20181126T022319_N0207_R103_T51PWM_20181126T050025.SAFE"

>>> # Create the reader object and open satellite data
>>> eoreader = Reader()

>>>  # The Reader will recognize the satellite type from its structure
>>> s2_prod = eoreader.open(s2_path)
```



```
>>> # Sentinel-1 GRD path
>>> s1_path = "S1B_EW_GRDM_1SDH_20200422T080459_20200422T080559_021254_028559_784D.zip"

>>> # Create the reader object and open satellite data
>>> eoreader = Reader()

>>> # The Reader will recognize the satellite type from its structure
>>> s1_prod = eoreader.open(s1_path)
```

Load and stack bands

Loading and stacking bands

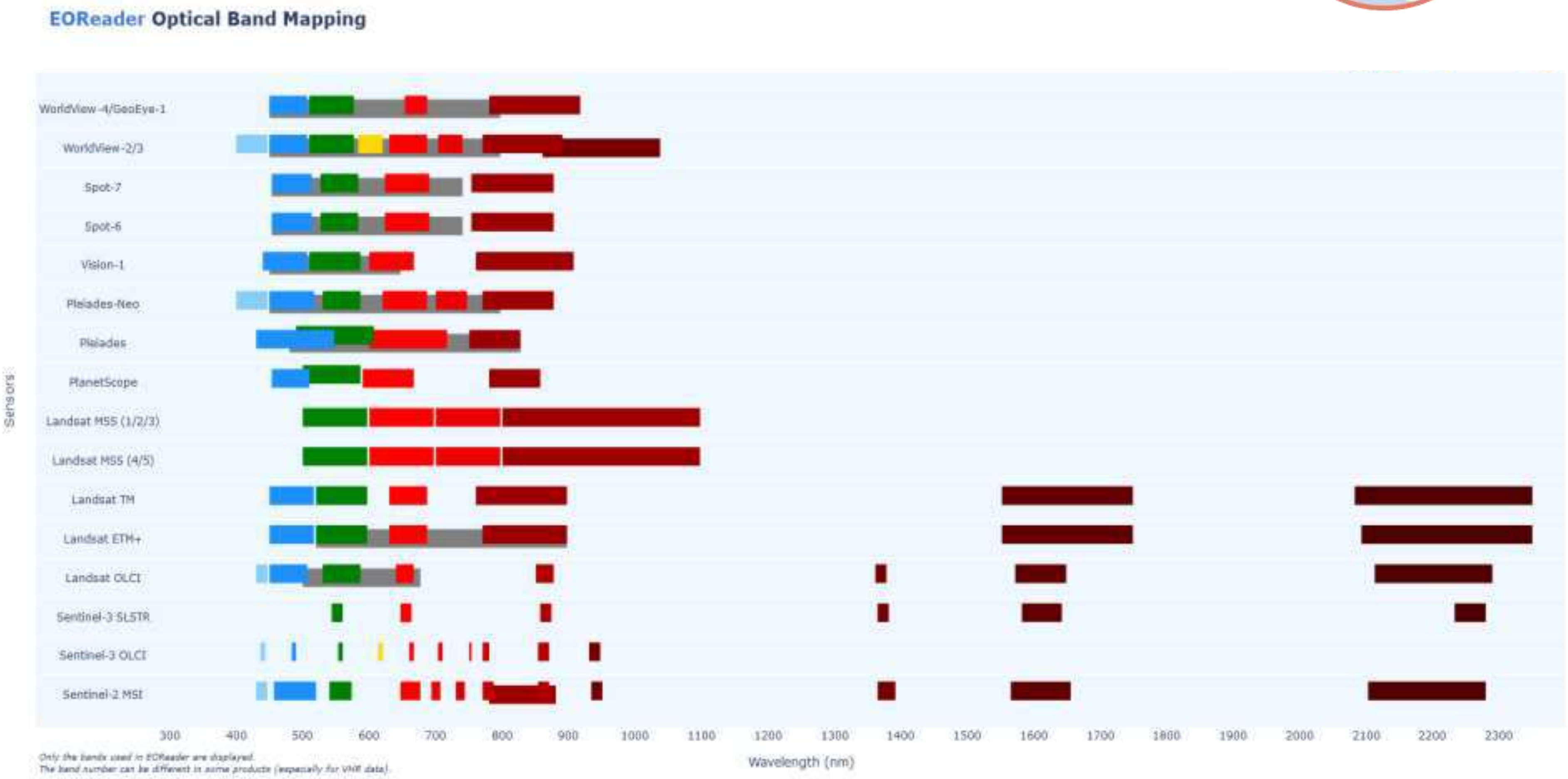


Optical bands

- Available **optical** bands
  - **Satellite** bands: `RED, NIR, SWIR, PAN, …`
  - **Optical** Index: `NDVI, NDWI, …`
  - **DEM** bands: `DEM, SLOPE, HILLSHADE`
  - **Cloud** bands (if existing): `CLOUDS, CIRRUS, SHADOWS, …`
- Bands specification
  - Always **orthorectified**
  - Always **projected** in UTM
  - Possibility to remove defective pixels, nodata set by default

Optical Band Mapping

- **Band mapping** between optical sensors



EOReader Optical Band Mapping

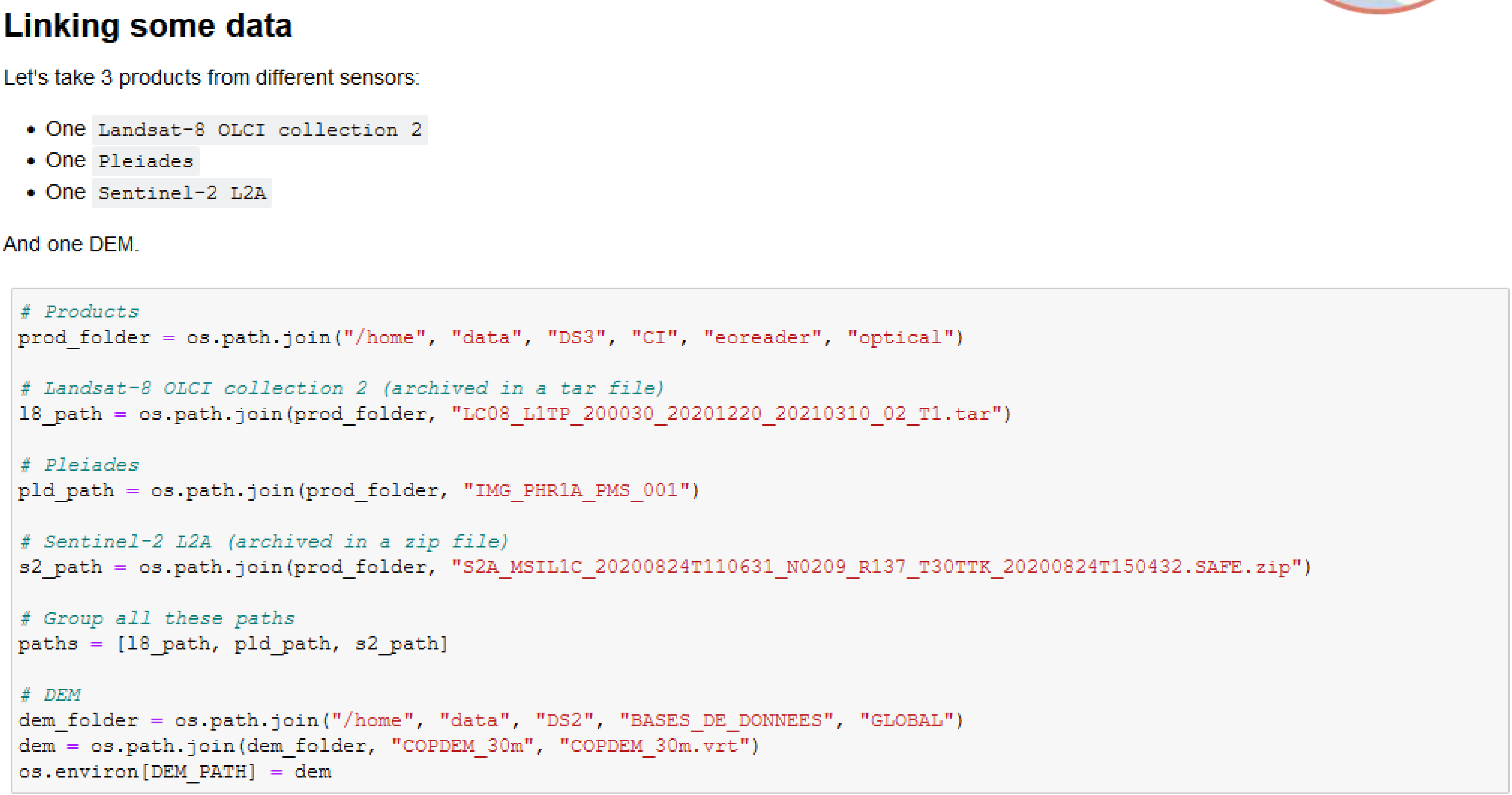
Only the bands used in EOReader are displayed.
The band number can be different in some products (especially for VHR data).

SAR bands

- Available **SAR** bands
  - **Satellite** bands: `VV, HH, HV, VH`
  - **Despeckled** bands: `VV_DSPK, HH_DSPK, HV_DSPK, VH_DSPK`
  - **DEM** bands (except hillshade): `DEM, SLOPE`
- Bands specification
  - Always **orthorectified**
  - Always **projected** in UTM
  - Nodata set by default

Load and stack bands

Example



Linking some data

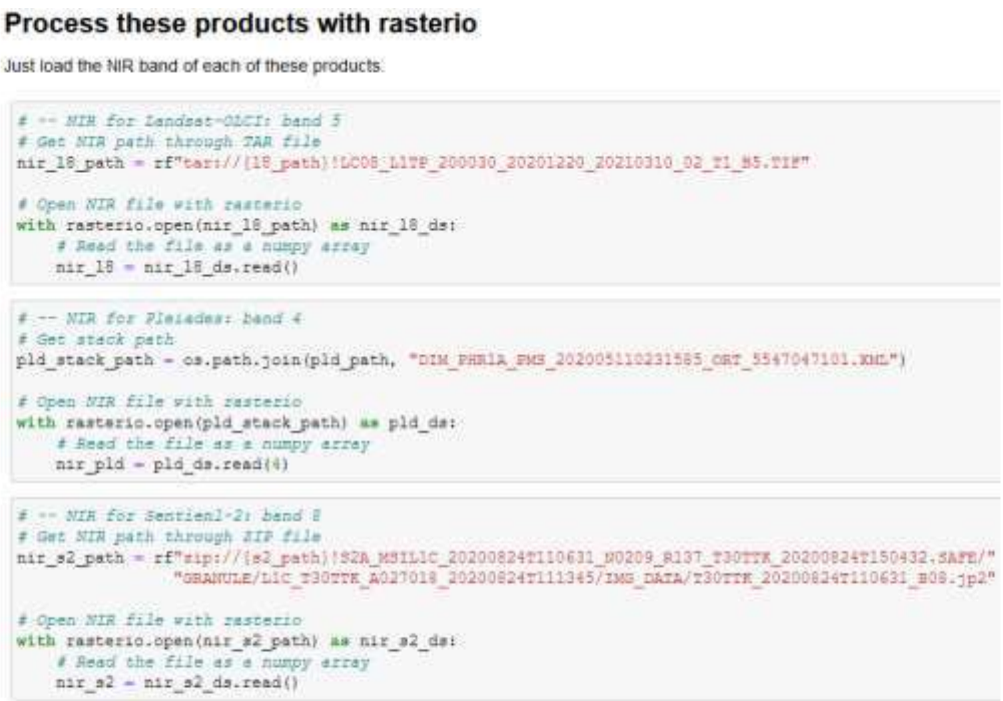
Let's take 3 products from different sensors:

- One `Landsat-8 OLCI collection 2`
- One `Pleiades`
- One `Sentinel-2 L2A`

And one DEM.

```python
# Products
prod_folder = os.path.join("/home", "data", "DS3", "CI", "eoreader", "optical")

# Landsat-8 OLCI collection 2 (archived in a tar file)
l8_path = os.path.join(prod_folder, "LC08_L1TP_200030_20201220_20210310_02_T1.tar")

# Pleiades
pld_path = os.path.join(prod_folder, "IMG_PHR1A_PMS_001")

# Sentinel-2 L2A (archived in a zip file)
s2_path = os.path.join(prod_folder, "S2A_MSIL1C_20200824T110631_N0209_R137_T30TTK_20200824T150432.SAFE.zip")

# Group all these paths
paths = [l8_path, pld_path, s2_path]

# DEM
dem_folder = os.path.join("/home", "data", "DS2", "BASES_DE_DONNEES", "GLOBAL")
dem = os.path.join(dem_folder, "COPDEM_30m", "COPDEM_30m.vrt")
os.environ[DEM_PATH] = dem
```

Load and stack bands



Process these products with rasterio

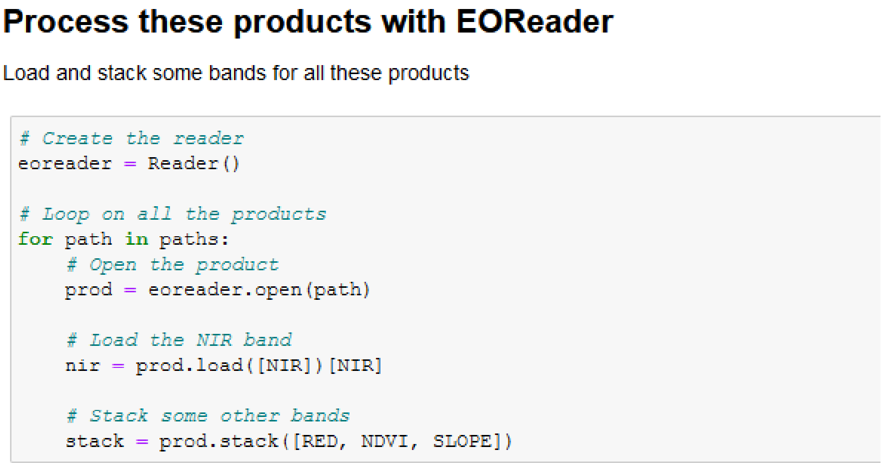
Just load the NIR band of each of these products.

```
# -- NIR for Landsat-OLCI: band 5
# Get NIR path through TAR file
nir_l8_path = rf"tar://{l8_path}!LC08_L1TP_200030_20201220_20210310_02_T1_B5.TIF"

# Open NIR file with rasterio
with rasterio.open(nir_l8_path) as nir_l8_ds:
    # Read the file as a numpy array
    nir_l8 = nir_l8_ds.read()
```

```
# -- NIR for Pleiades: band 4
# Get stack path
pld_stack_path = os.path.join(pld_path, "DIM_PHR1A_PMS_202005110231585_ORT_5547047101.XML")

# Open NIR file with rasterio
with rasterio.open(pld_stack_path) as pld_ds:
    # Read the file as a numpy array
    nir_pld = pld_ds.read(4)
```

```
# -- NIR for Sentienl-2: band 8
# Get NIR path through ZIP file
nir_s2_path = rf"zip://{s2_path}!S2A_MSIL1C_20200824T110631_N0209_R137_T30TTK_20200824T150432.SAFE/"
                "GRANULE/L1C_T30TTK_A027018_20200824T111345/IMG_DATA/T30TTK_20200824T110631_B08.jp2"

# Open NIR file with rasterio
with rasterio.open(nir_s2_path) as nir_s2_ds:
    # Read the file as a numpy array
    nir_s2 = nir_s2_ds.read()
```

- complicated paths for archived data
- nodata not set if not saved into the file
- bands not scaled (still in uint16, need to read the metadata in order to scale them)
- need to use other lib to read as xarray (rioxarray)
- not necessarily orthorectified or projected
- need to adapt this piece of code for each band



Process these products with EOReader

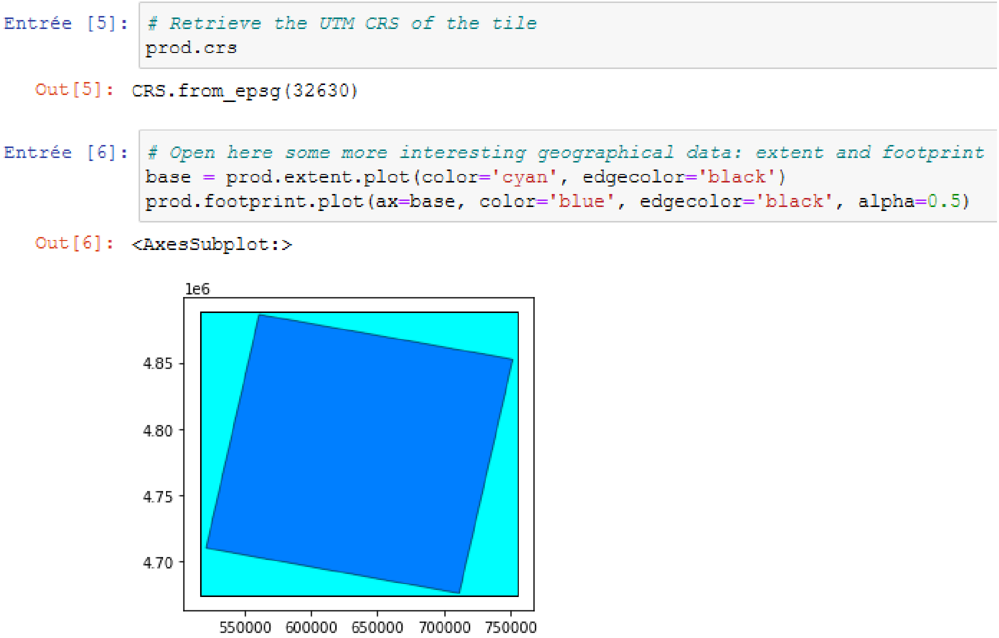
Load and stack some bands for all these products

```
# Create the reader
eoreader = Reader()

# Loop on all the products
for path in paths:
    # Open the product
    prod = eoreader.open(path)

    # Load the NIR band
    nir = prod.load([NIR])[NIR]

    # Stack some other bands
    stack = prod.stack([RED, NDVI, SLOPE])
```

- read both archived and extracted data the same way
- masked and scaled bands, converted to float32
- always orthorectified or projected in UTM
- loaded as xarray.DataArray
- same logic to load every band, whether a SLOPE, and index or an optical band is asked by the user
- manage stacks automatically
- load collocated bands

Other features

- **Geographical** data (always projected in UTM)
  - CRS
  - Extent
  - Footprint



```
Entrée [5]: # Retrieve the UTM CRS of the tile
            prod.crs

  Out[5]: CRS.from_epsg(32630)

Entrée [6]: # Open here some more interesting geographical data: extent and footprint
            base = prod.extent.plot(color='cyan', edgecolor='black')
            prod.footprint.plot(ax=base, color='blue', edgecolor='black', alpha=0.5)

  Out[6]: <AxesSubplot:>
```

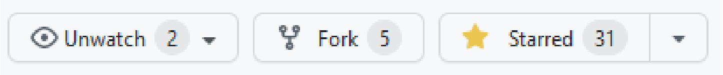
- **Solar angles**
  - Azimuth angle (mean)
  - Zenith angle (mean)
- Read **metadata**
  - As a `lxml Element` and a dictionary containing the namespaces

About the project

About the project

EOReader highlights

- First lines: **2.5 years old**
- Creation of the library: **05/03/2021**
- Release in open source: **28/04/2021**
- Version: **0.13.0**
- **700** commits, **20k** lines of python code
- Used **daily** at SERTIT in a lot of **production** tools
- Growing interest in the community



About the project

Why Python?

- Used by a **lot** of people, useful **scientific** libraries exist in **open source**, very **easy** to learn
- We code in Python in SERTIT

Why Open source?

- **Promote** a hidden code brick
- A lot of open source libraries are used in SERTIT, so we wanted to **contribute** back
- ***Soft power*** to show that we code here at SERTIT

About the project

EOReader standards

- **Lint** (`pep8` through `black`, `flake8`, `isort`, `pre-commit`)
- **Clear and complete documentation** (Readme, docstrings, code comments, API and tutorials)
- **Code Coverage** > 85% (currently 95%)
- **License** Apache 2.0

About the project

EOReader main dependencies

- Main **python** libraries
  - `rioxarray (raster + rasterio)`
  - `geopandas`
  - `lxml`
- **Other softwares**
  - ESA SNAP

About the project

Tools

- IDE: **Pycharm**
- Versioning tool: **Git**
- Package management system: **Pip** and **conda**
- Stored in **Github**, mirrored on an internal instance of **Gitlab**
- Documentation (scientific, tutorials and API) on **readthedocs**

Testing

- **Thematic validation** of the results by SERTIT experts
- **CI** on Gitlab (on commits and weekly)

About the project

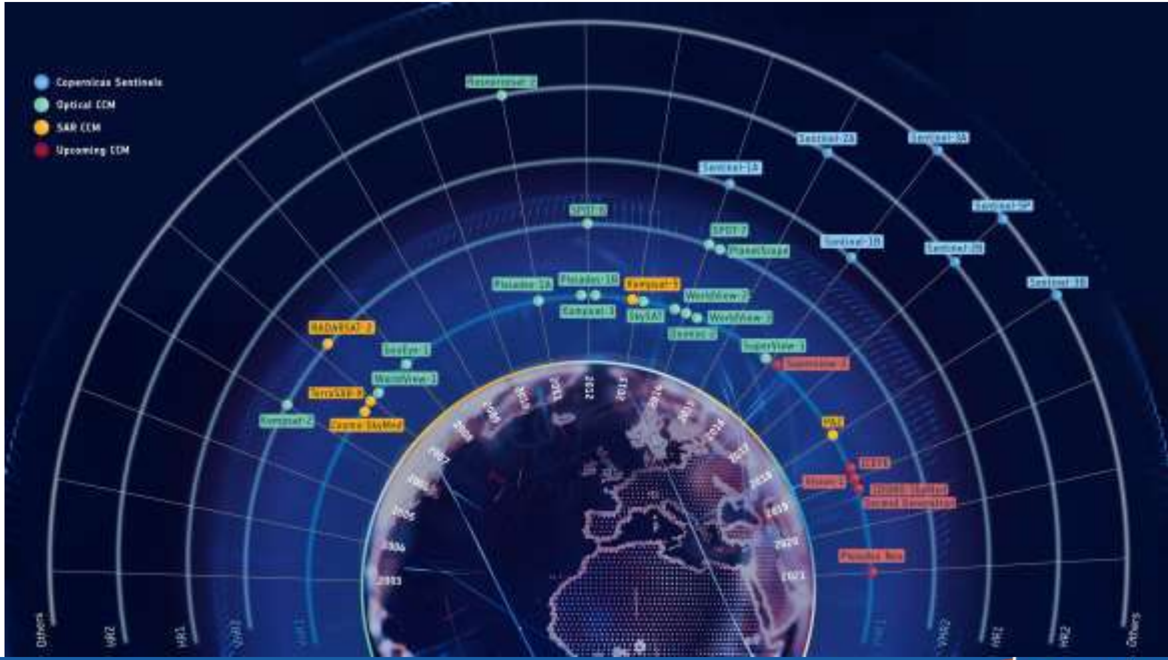
EOReader challenges

- **Match** community and corporates **needs** and **priorities**
- Manage **multi-platform** code (Linux and Windows)
- Master **dependencies** (security and exponential growth)

About the project

EOReader’s future

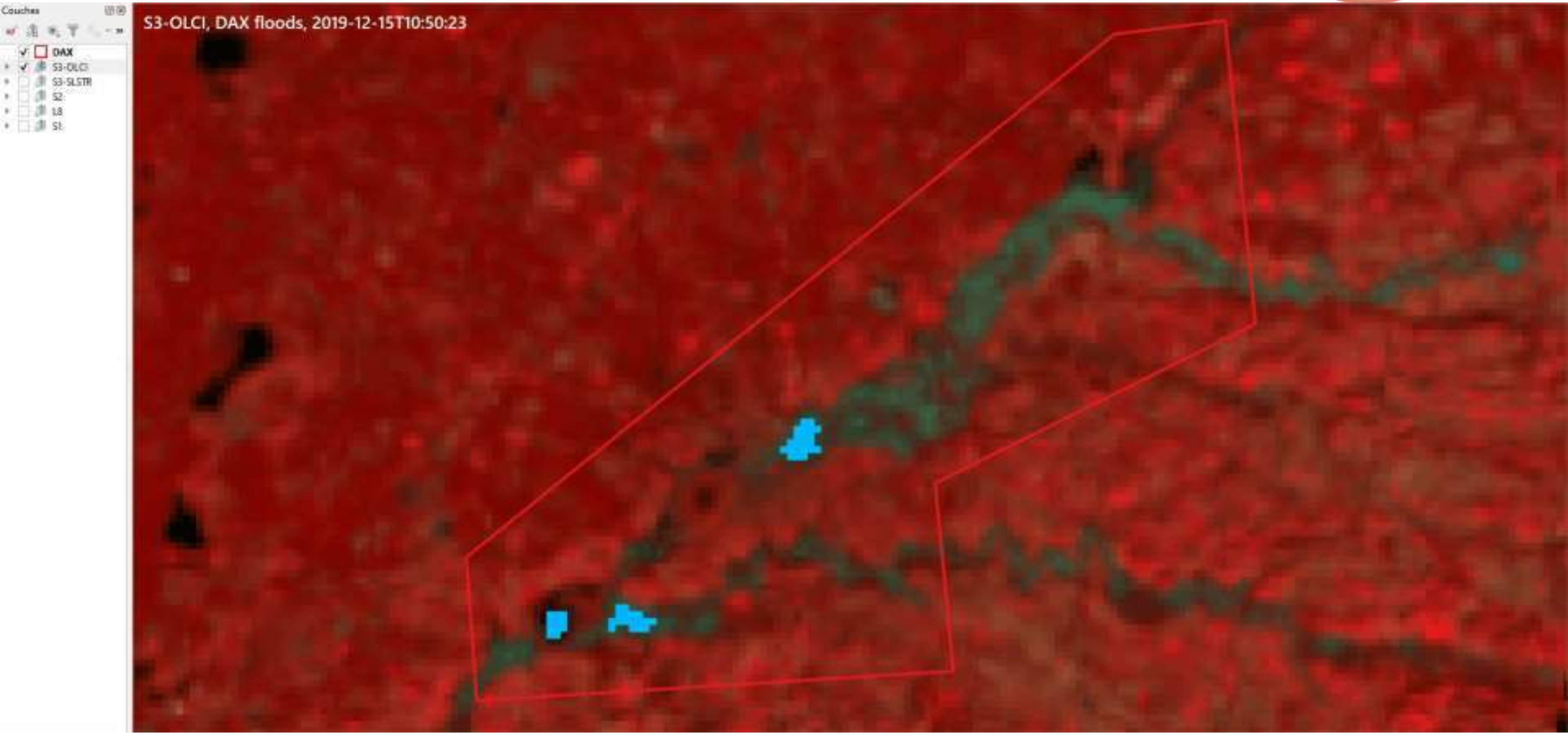
- Get rid of **big non python external** tools (such as ESA SNAP)
- Make sure the code is **optimized** (speed, memory consumption)
- Implement all of used **CEMS sensors**



Thank you for your attention

EOReader

Flood delineations

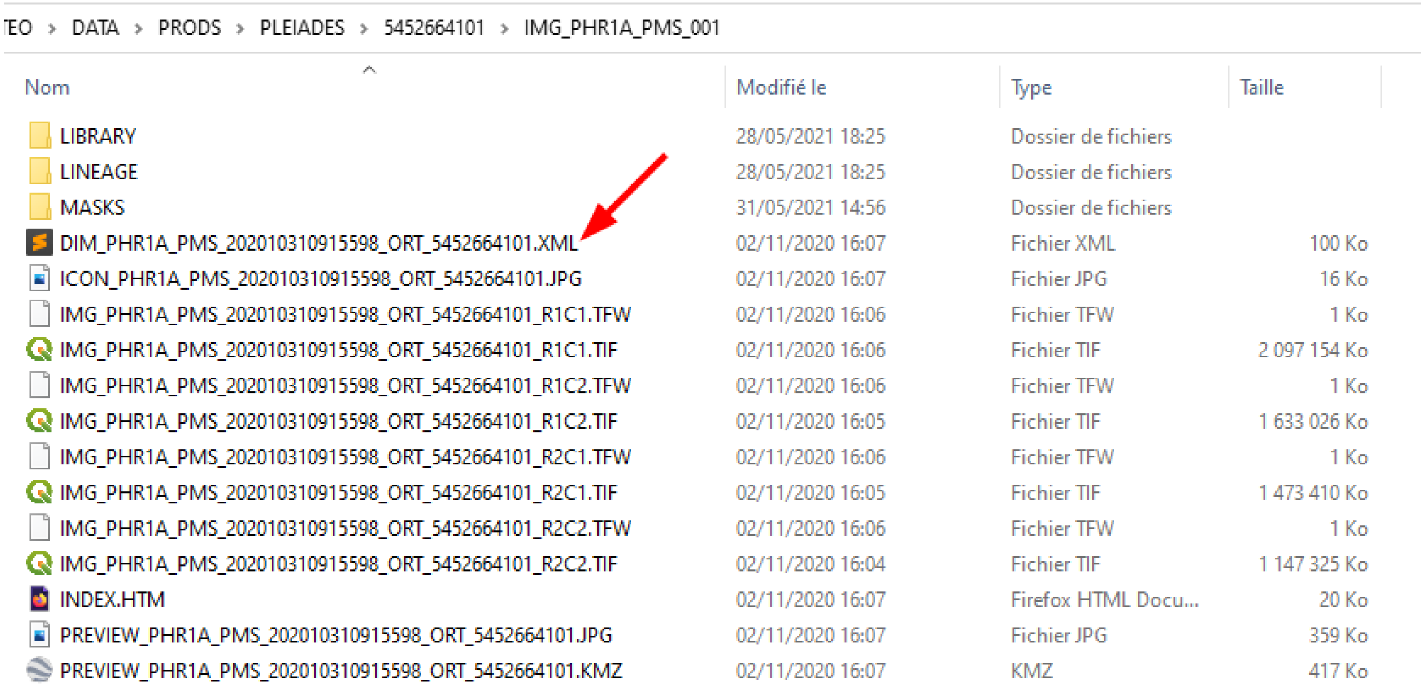


S3-OLCI, DAX floods, 2019-12-15T10:50:23

EOReader

Satellite differences: **Pléiades**

- Stacked, tiled, may need orthorectification
- Red = band n°1 (inside the stack)



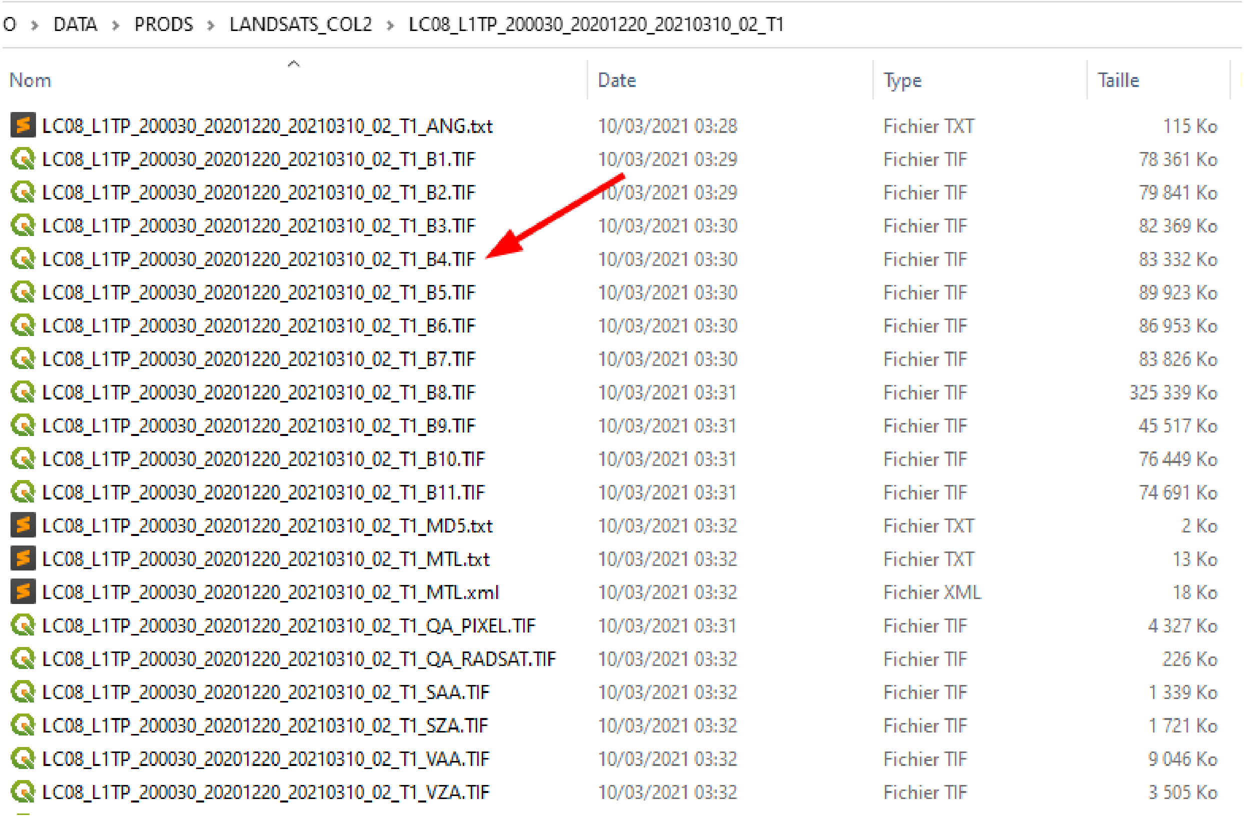
EO › DATA › PRODS › PLEIADES › 5452664101 › IMG_PHR1A_PMS_001

Nom	Modifié le	Type	Taille
LIBRARY	28/05/2021 18:25	Dossier de fichiers	
LINEAGE	28/05/2021 18:25	Dossier de fichiers	
MASKS	31/05/2021 14:56	Dossier de fichiers	
DIM_PHR1A_PMS_202010310915598_ORT_5452664101.XML	02/11/2020 16:07	Fichier XML	100 Ko
ICON_PHR1A_PMS_202010310915598_ORT_5452664101.JPG	02/11/2020 16:07	Fichier JPG	16 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R1C1.TFW	02/11/2020 16:06	Fichier TFW	1 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R1C1.TIF	02/11/2020 16:06	Fichier TIF	2 097 154 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R1C2.TFW	02/11/2020 16:06	Fichier TFW	1 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R1C2.TIF	02/11/2020 16:05	Fichier TIF	1 633 026 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R2C1.TFW	02/11/2020 16:06	Fichier TFW	1 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R2C1.TIF	02/11/2020 16:05	Fichier TIF	1 473 410 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R2C2.TFW	02/11/2020 16:06	Fichier TFW	1 Ko
IMG_PHR1A_PMS_202010310915598_ORT_5452664101_R2C2.TIF	02/11/2020 16:04	Fichier TIF	1 147 325 Ko
INDEX.HTM	02/11/2020 16:07	Firefox HTML Docu...	20 Ko
PREVIEW_PHR1A_PMS_202010310915598_ORT_5452664101.JPG	02/11/2020 16:07	Fichier JPG	359 Ko
PREVIEW_PHR1A_PMS_202010310915598_ORT_5452664101.KMZ	02/11/2020 16:07	KMZ	417 Ko

EOReader

Satellite differences: **Landsat-8**

- 1 file per band
- Red = band n° 4



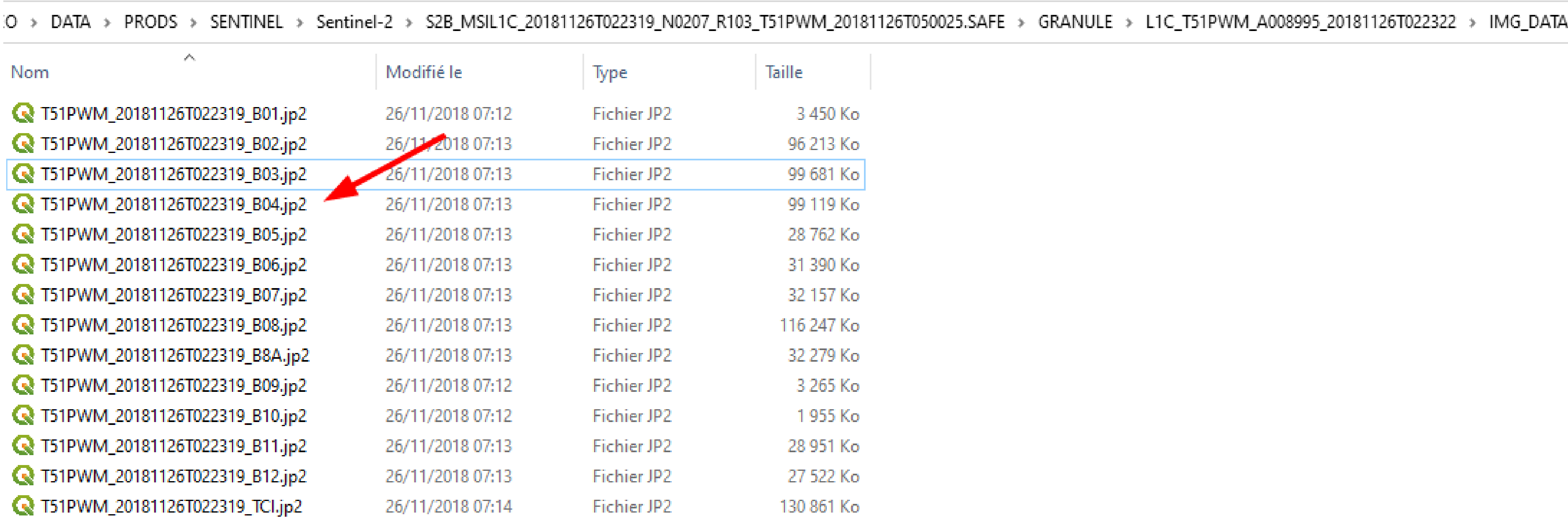
O › DATA › PRODS › LANDSATS_COL2 › LC08_L1TP_200030_20201220_20210310_02_T1

Nom	Date	Type	Taille
LC08_L1TP_200030_20201220_20210310_02_T1_ANG.txt	10/03/2021 03:28	Fichier TXT	115 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B1.TIF	10/03/2021 03:29	Fichier TIF	78 361 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B2.TIF	10/03/2021 03:29	Fichier TIF	79 841 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B3.TIF	10/03/2021 03:30	Fichier TIF	82 369 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B4.TIF	10/03/2021 03:30	Fichier TIF	83 332 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B5.TIF	10/03/2021 03:30	Fichier TIF	89 923 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B6.TIF	10/03/2021 03:30	Fichier TIF	86 953 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B7.TIF	10/03/2021 03:30	Fichier TIF	83 826 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B8.TIF	10/03/2021 03:31	Fichier TIF	325 339 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B9.TIF	10/03/2021 03:31	Fichier TIF	45 517 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B10.TIF	10/03/2021 03:31	Fichier TIF	76 449 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_B11.TIF	10/03/2021 03:31	Fichier TIF	74 691 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_MD5.txt	10/03/2021 03:32	Fichier TXT	2 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_MTL.txt	10/03/2021 03:32	Fichier TXT	13 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_MTL.xml	10/03/2021 03:32	Fichier XML	18 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_QA_PIXEL.TIF	10/03/2021 03:31	Fichier TIF	4 327 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_QA_RADSAT.TIF	10/03/2021 03:32	Fichier TIF	226 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_SAA.TIF	10/03/2021 03:32	Fichier TIF	1 339 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_SZA.TIF	10/03/2021 03:32	Fichier TIF	1 721 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_VAA.TIF	10/03/2021 03:32	Fichier TIF	9 046 Ko
LC08_L1TP_200030_20201220_20210310_02_T1_VZA.TIF	10/03/2021 03:32	Fichier TIF	3 505 Ko

EOReader

Satellite differences: **Sentinel-2**

- 1 file per band, very deep file tree, in JPEG-2000 format
- Red = band n° 4



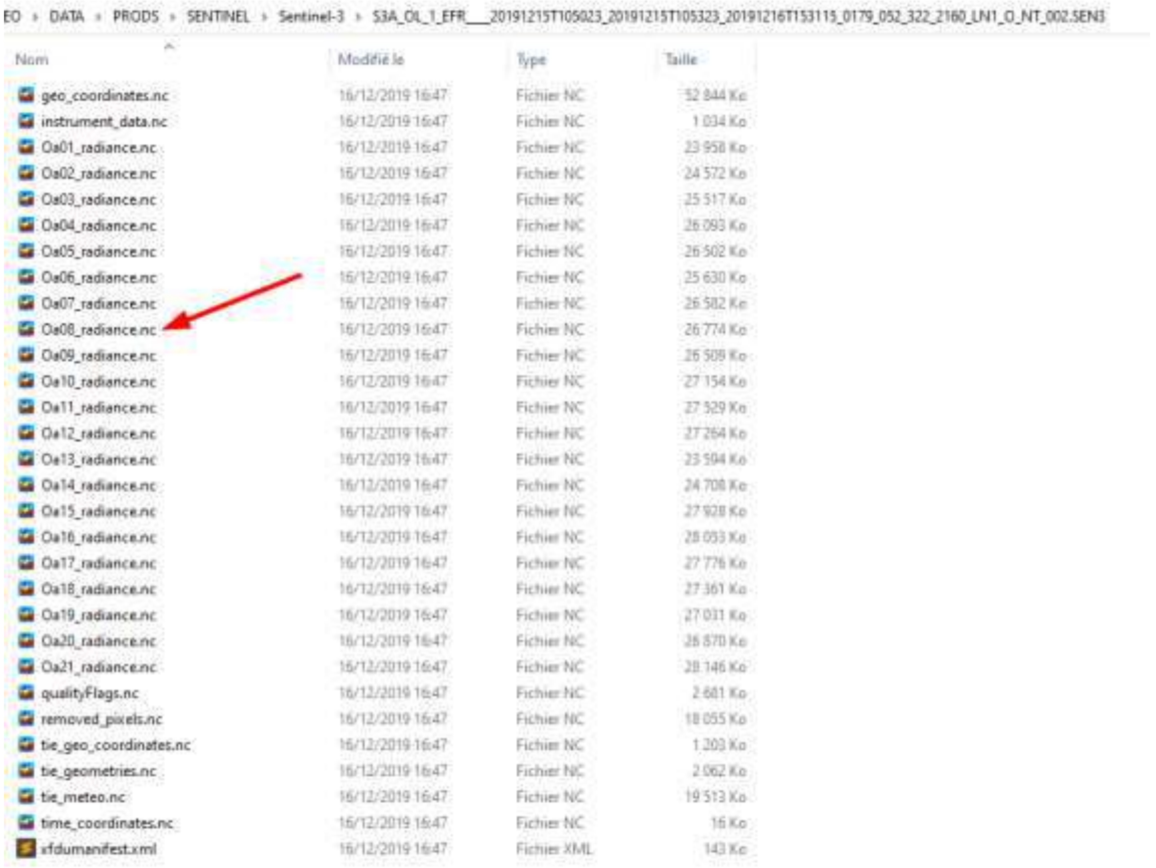
O › DATA › PRODS › SENTINEL › Sentinel-2 › S2B_MSIL1C_20181126T022319_N0207_R103_T51PWM_20181126T050025.SAFE › GRANULE › L1C_T51PWM_A008995_20181126T022322 › IMG_DATA

Nom	Modifié le	Type	Taille
T51PWM_20181126T022319_B01.jp2	26/11/2018 07:12	Fichier JP2	3 450 Ko
T51PWM_20181126T022319_B02.jp2	26/11/2018 07:13	Fichier JP2	96 213 Ko
T51PWM_20181126T022319_B03.jp2	26/11/2018 07:13	Fichier JP2	99 681 Ko
T51PWM_20181126T022319_B04.jp2	26/11/2018 07:13	Fichier JP2	99 119 Ko
T51PWM_20181126T022319_B05.jp2	26/11/2018 07:13	Fichier JP2	28 762 Ko
T51PWM_20181126T022319_B06.jp2	26/11/2018 07:13	Fichier JP2	31 390 Ko
T51PWM_20181126T022319_B07.jp2	26/11/2018 07:13	Fichier JP2	32 157 Ko
T51PWM_20181126T022319_B08.jp2	26/11/2018 07:13	Fichier JP2	116 247 Ko
T51PWM_20181126T022319_B8A.jp2	26/11/2018 07:13	Fichier JP2	32 279 Ko
T51PWM_20181126T022319_B09.jp2	26/11/2018 07:12	Fichier JP2	3 265 Ko
T51PWM_20181126T022319_B10.jp2	26/11/2018 07:12	Fichier JP2	1 955 Ko
T51PWM_20181126T022319_B11.jp2	26/11/2018 07:13	Fichier JP2	28 951 Ko
T51PWM_20181126T022319_B12.jp2	26/11/2018 07:13	Fichier JP2	27 522 Ko
T51PWM_20181126T022319_TCI.jp2	26/11/2018 07:14	Fichier JP2	130 861 Ko

EOReader

Satellite differences: **Sentinel-3**

- 1 file per band, needs geocoding, in NetCDF format
- Red = band n° 8



EO › DATA › PRODS › SENTINEL › Sentinel-3 › S3A_OL_1_EFR____20191215T105023_20191215T105323_20191216T153115_0179_052_322_2160_LN1_O_NT_002.SEN3

Nom	Modifié le	Type	Taille
geo_coordinates.nc	16/12/2019 16:47	Fichier NC	52 844 Ko
instrument_data.nc	16/12/2019 16:47	Fichier NC	1 034 Ko
Oa01_radiance.nc	16/12/2019 16:47	Fichier NC	23 958 Ko
Oa02_radiance.nc	16/12/2019 16:47	Fichier NC	24 572 Ko
Oa03_radiance.nc	16/12/2019 16:47	Fichier NC	25 517 Ko
Oa04_radiance.nc	16/12/2019 16:47	Fichier NC	26 093 Ko
Oa05_radiance.nc	16/12/2019 16:47	Fichier NC	26 502 Ko
Oa06_radiance.nc	16/12/2019 16:47	Fichier NC	25 630 Ko
Oa07_radiance.nc	16/12/2019 16:47	Fichier NC	26 582 Ko
Oa08_radiance.nc	16/12/2019 16:47	Fichier NC	26 774 Ko
Oa09_radiance.nc	16/12/2019 16:47	Fichier NC	26 509 Ko
Oa10_radiance.nc	16/12/2019 16:47	Fichier NC	27 154 Ko
Oa11_radiance.nc	16/12/2019 16:47	Fichier NC	27 529 Ko
Oa12_radiance.nc	16/12/2019 16:47	Fichier NC	27 264 Ko
Oa13_radiance.nc	16/12/2019 16:47	Fichier NC	23 594 Ko
Oa14_radiance.nc	16/12/2019 16:47	Fichier NC	24 708 Ko
Oa15_radiance.nc	16/12/2019 16:47	Fichier NC	27 928 Ko
Oa16_radiance.nc	16/12/2019 16:47	Fichier NC	28 053 Ko
Oa17_radiance.nc	16/12/2019 16:47	Fichier NC	27 776 Ko
Oa18_radiance.nc	16/12/2019 16:47	Fichier NC	27 361 Ko
Oa19_radiance.nc	16/12/2019 16:47	Fichier NC	27 031 Ko
Oa20_radiance.nc	16/12/2019 16:47	Fichier NC	28 870 Ko
Oa21_radiance.nc	16/12/2019 16:47	Fichier NC	28 146 Ko
qualityFlags.nc	16/12/2019 16:47	Fichier NC	2 681 Ko
removed_pixels.nc	16/12/2019 16:47	Fichier NC	18 055 Ko
tie_geo_coordinates.nc	16/12/2019 16:47	Fichier NC	1 203 Ko
tie_geometries.nc	16/12/2019 16:47	Fichier NC	2 062 Ko
tie_meteo.nc	16/12/2019 16:47	Fichier NC	19 513 Ko
time_coordinates.nc	16/12/2019 16:47	Fichier NC	16 Ko
xfdumanifest.xml	16/12/2019 16:47	Fichier XML	143 Ko