



Image processing for Earth observation

How to access
geo-information?

Gaston Lenczner

September 2024

Download data in remote sensing

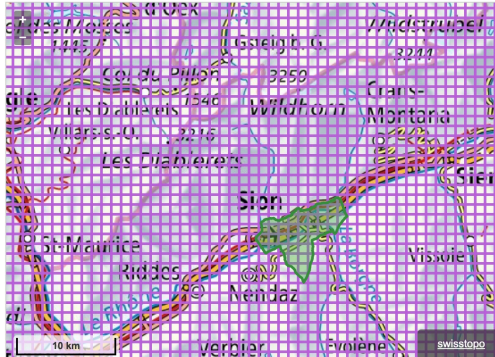
- As in all fields of data science, a key step is to obtain the data
- Multiple platforms and sources provide aerial and satellite data
- Some are free, some are not, some provide a limited and/or conditional free offer
- For manual labelling of geospatial data and local visualization: QGIS (<https://qgis.org/>)

- RGB orthophotos at 10cm (or 25 cm depending on the area) of Switzerland
- One release per year since 2017: Free
- Upon request before (started in 1946)
- Digital Elevation Model available
- Maps: potential machine learning labels
- Extremely easy-to-download

Selection mode

Selection by municipality

Q Sion (VS)



Options

Format: Cloud Optimized GeoTIFF

Resolution [meters]: 0.1

Coordinate system: LV95

Time status: ☒ Current

Published since: ☐

Published before: ☐

Search

Copernicus browser – Overall

<https://browser.dataspace.copernicus.eu/>

- Free to use and publicly owned tool (ESA Project)
- Visualise, compare and analyse and download data
- Limited amount of freely available processing data
- Sentinel, Landsat, MODIS, Probas-V and GIBS products available
- Easily visualize pre-computed layers (NDVI, NDWI, ...)
- Manual exporting method not suitable for large data sets

Copernicus browser – User interface

The image displays the Copernicus browser user interface. On the left, a sidebar contains controls for visualization, data collections, and layers. On the right, a large satellite image of a mountainous region is shown.

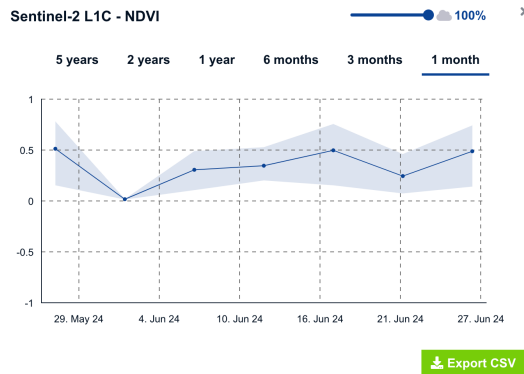
Left Sidebar:

- VISUALIZE** (highlighted in green)
- SEARCH**
- Date:** A date range selector showing "2024-08-31" with navigation arrows and a "30%" zoom level.
- Satellite source:** A dropdown menu showing "Sentinel-2" with a list of data collections: "Sentinel-2 L1C" and "Sentinel-2 L2A" (checked).
- Product:** A list of visualization products: "True color" (Based on bands B4, B3, B2), "False color" (Based on bands B8, B4, B3), "Highlight Optimized Natural Color" (Enhanced natural color visualization), "NDVI" (Based on a combination of bands (B8 - B4)/(B8 + B4)), "False color (urban)" (Based on bands B12, B11, B4), "Moisture index" (Based on a combination of bands (B8A - B11)/(B8A + B11)), "SWIR" (Based on bands B12, B8A, B4), and "NDWI".

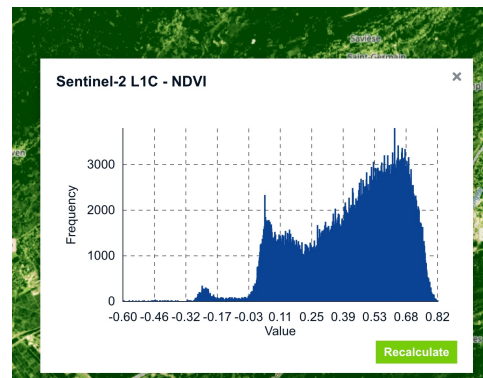
Right Panel:

- A large satellite image of a mountainous region, showing a river valley and surrounding hills. The image is labeled with various geographical features and coordinates.

- Draw or import an region of interest
- Get the time serie of an index in a selected area with a single button
- Get the histogram of the selected area with a single button



Time serie



Histogram

×

Basic Analytical High-res print

Image download

Image format: TIFF (16-bit) ▼

Image resolution: CUSTOM ▼
362 x 341 px

Resolution X (m/px): 10

Resolution Y (m/px): 10

Coordinate system: WGS 84 (EPSG:4326) ▼

Add dataMask band to raw layers ⓘ ☒

Layers:

Visualized

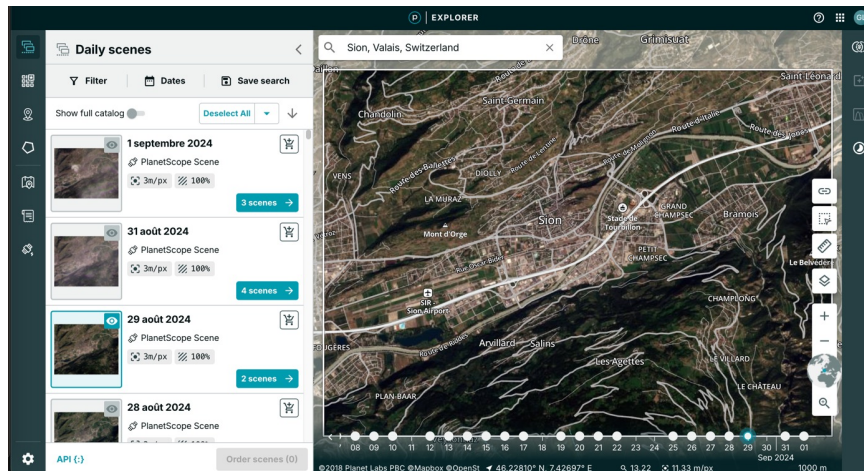
☐ True color
☐ False color
☐ Highlight Optimized Natural Color
☐ NDVI
☐ False color (urban)
☐ Moisture index
☐ SWIR
☐ NDWI
☐ NDSI

Raw

☒ B01
☒ B02
☒ B03
☒ B04
☒ B05
☐ B06
☐ B07
☐ B08
☐ B8A
☐ B09
☐ B10
☐ B11
☐ B12

 Download

- Education and Research Standard Plan: free to download 5000km²/month.
- High resolution imagery (3m)
- Manual exporting method not suitable for large data sets
- User friendly interface to specify the area of interest, the dates and deal with parameters like cloud coverage



Google Earth Engine – What is it?

<https://code.earthengine.google.com>

- **Cloud-based platform** for processing vast amounts of satellite imagery and geospatial data.
- **Programming interface:** Uses JavaScript and Python APIs
- **Large-scale data processing:** Can analyze large amounts of data without needing local infrastructure.
- **Support machine learning**

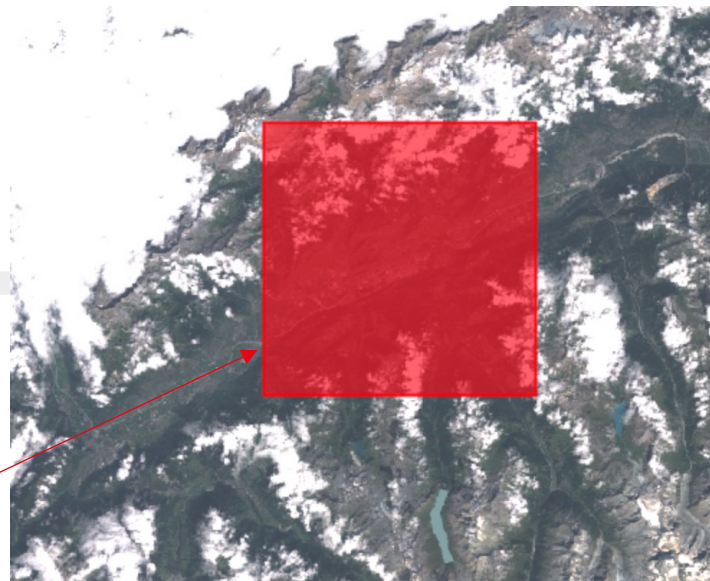
- Large catalog of remote sensing datasets
 - Imagery (e.g. Sentinel, Landsat)
 - Climate (e.g. ERA5)
 - Surface temperature (e.g. MODIS)
 - Digital Elevation Models (e.g. SRTM)
 - Land Cover (e.g. ESA World Cover)
- Real-time data availability: Many datasets are updated regularly.
- Web-based Javascript interface and Python API
 - Easy to share results and scripts with others
- Free for non-commercial use
- Usable to download big amounts of data with coding scripts

Google Earth Engine - Cons

- Requires a Google account.
- Requires some coding knowledge.
- **Impractical to export data outside the platform:**
 - Through Google drive with the Javascript API
 - Natively supported, easy-to-do
 - Manual method not suitable for large data sets
 - A file size limit
 - Through Python API
 - A file size limit that can be bypassed by splitting the data into smaller parts
- Commercial use restrictions (not our concern)
- Google can change the way it works at any time.
- Need to be careful when downloading a new dataset:
 - SAR data are log transformed, normalized and thresholded.

```
1
2 var sion = ee.Geometry.Point([7.3606, 46.2331]); // Define the location of Sion, Switzerland
3 var region = sion.buffer(10000).bounds(); // 10 km buffer around Sion as a bounding box
4
5 // Load S2 image collection
6 var s2 = ee.ImageCollection('COPERNICUS/S2').filterBounds(sion).filterDate('2024-07-01', '2024-08-31');
7 // Take most recent image
8 var s2_first = s2.sort('system:time_start', false).first()
9 // Select the RGB bands (Red, Green, Blue)
10 var rgb = s2_first.select(['B4', 'B3', 'B2']);
11
12 // Center the map on Sion and display the RGB image
13 Map.centerObject(region, 10);
14 Map.addLayer(rgb, {min: 0, max: 5000}, 'S2 RGB');
15
16 // Highlight the region on the map
17 Map.addLayer(region, {color: 'red'}, 'Region of Interest');
18
19 // Export the image to Google Drive
20 Export.image.toDrive({
21   image: rgb,
22   region: region,
23   description: 'Sion_S2_RGB_Summer2024',
24   scale: 10, // S2 has 10m resolution for RGB bands
25 });
```

```
1
2 var sion = ee.Geometry.Point([7.3606, 46.2331]); // Define the location of Sion, Switzerland
3 var region = sion.buffer(10000).bounds(); // 10 km buffer around Sion as a bounding box
4
5 // Load S2 image collection
6 var s2 = ee.ImageCollection('COPERNICUS/S2').filterBounds(sion).filterDate('2024-07-01', '2024-08-31');
7 // Take most recent image
8 var s2_first = s2.sort('system:time_start', false).first()
9 // Select the RGB bands (Red, Green, Blue)
10 var rgb = s2_first.select(['B4', 'B3', 'B2']);
11
12 // Center the map on Sion and display the RGB image
13 Map.centerObject(region, 10);
14 Map.addLayer(rgb, {min: 0, max: 5000}, 'S2 RGB');
15
16 // Highlight the region on the map
17 Map.addLayer(region, {color: 'red'}, 'Region of Interest');
18
19 // Export the image to Google Drive
20 Export.image.toDrive({
21   image: rgb,
22   region: region,
23   description: 'Sion_S2_RGB_Summer2024',
24   scale: 10, // S2 has 10m resolution for RGB bands
25 });
```



Target area to download

Possible to add customized pre processing:
here, we compute the median over July and August and remove clouded images

```

1 // Mask clouds
2 var MAX_CLOUD_PROBABILITY = 20;
3 function maskClouds(s2, s2Clouds){
4   var s2EdgesMasked = s2.map(function(img){
5     return img.updateMask(img.select('B8A').mask().updateMask(img.select('B9').mask()))
6   })
7   var s2SrWithCloudMask = ee.Join.saveFirst('cloud_mask').apply({
8     primary: s2EdgesMasked, secondary: s2Clouds,
9     condition: ee.Filter.equals({leftField: 'system:index', rightField: 'system:index'})
10  })
11  var s2CloudMasked = ee.ImageCollection(s2SrWithCloudMask).map(function(img){
12    return img.updateMask(ee.Image(img.get('cloud_mask')).select('probability').lt(MAX_CLOUD_PROBABILITY))
13  })
14  return s2CloudMasked
15 }
16
17
18 var sion = ee.Geometry.Point([7.3606, 46.2331]); // Define the location of Sion, Switzerland
19 var region = sion.buffer(10000).bounds(); // 10 km buffer around Sion as a bounding box
20
21 // Load S2 image collection
22 var s2 = ee.ImageCollection('COPERNICUS/S2').filterBounds(sion).filterDate('2024-07-01', '2024-08-31');
23 // Load associated cloud probability
24 var s2Clouds = ee.ImageCollection('COPERNICUS/S2_CLOUD_PROBABILITY').filterDate('2024-07-01', '2024-08-31').filterBounds(sion);
25 // Take median over all images that do not contain clouds
26 var s2_median = maskClouds(s2, s2Clouds).median()
27 // Select the RGB bands (Red, Green, Blue)
28 var rgb = s2_median.select(['B4', 'B3', 'B2']);
29
30 // Center the map on Sion and display the RGB image
31 Map.centerObject(region, 10);
32 Map.addLayer(rgb, {min: 0, max: 5000}, 'S2 RGB');
```

Google Earth Engine – Visualize data



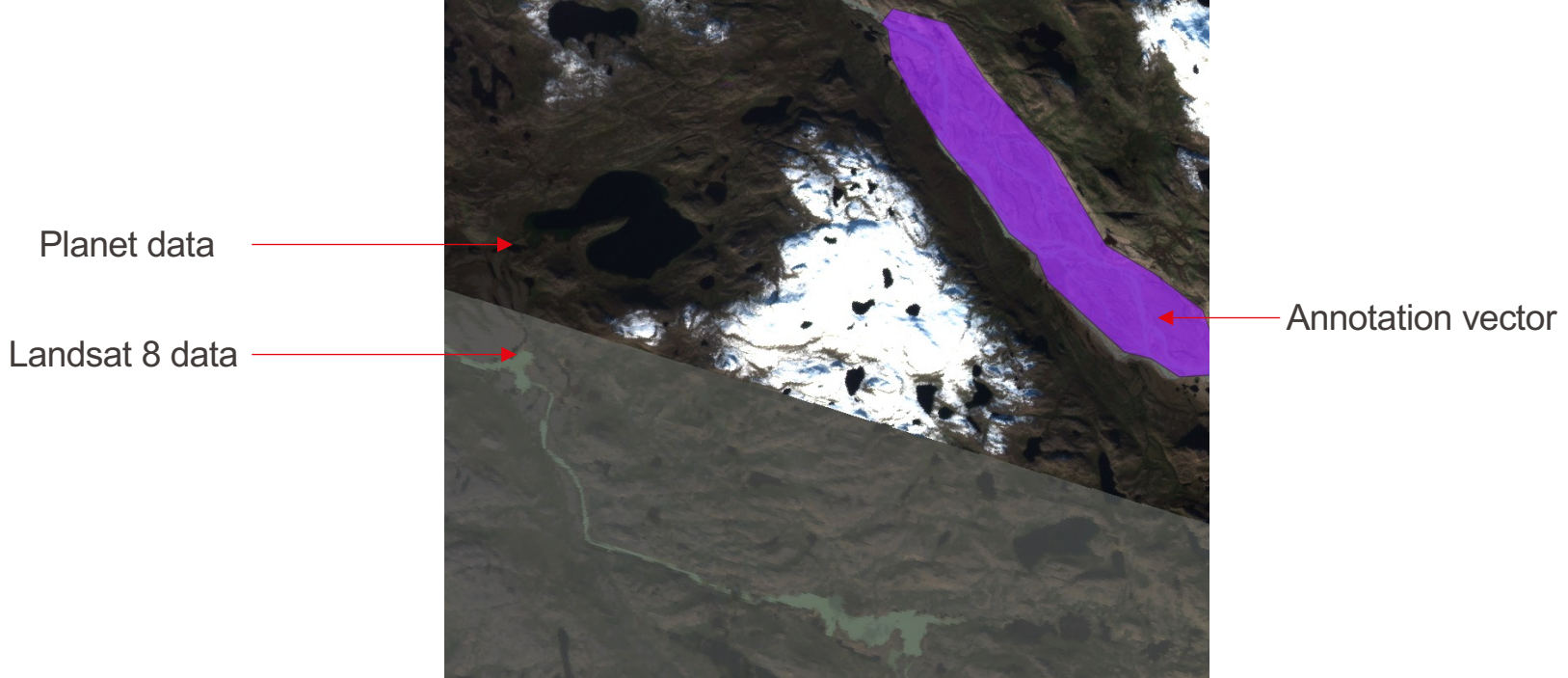
S2 image without preprocessing on one date



Median S2 image with cloud masking
over July and August 2024



QGIS – Visualize and annotate locally



Landsat RGB imagery overlayed with Planet RGB imagery and outwash plain annotation in Greenland

Take-home message

- No *easy* and *free* way to download large amounts of remote sensing images
 - Through Planet explore and Copernicus browser: Good for visualization and downloading small datasets.
 - Through Google Earth Engine: Many datasets and possible large files exports but with some coding tricks.
 - Swisstopo for Switzerland
- Labels can be available depending on the country policy
 - Switzerland: Swisstopo, France: IGN, ...
- Many other platforms exist (PEPS by CNES, USGS.gov, ...)
- QGIS to work locally with GIS data.
- Dedicated geospatial Python libraries
 - Rasterio, fiona, shapely, geopandas, ...

Bonus: Find your name in Landsat

<https://landsat.gsfc.nasa.gov/apps/YourNameInLandsat-main/>

Your Name In Landsat

Type in your name to see it spelled out in Landsat imagery of Earth!

