

Create GEOTIFF I/O template in Python.

The folder “code” contains two files:

1. a Jupyter Notebook containing the function and the example.
2. a Python file containing the function.

Required libraries:

- NumPy
- Rasterio

The example has the purpose of creating a GeoTIFF file (correctly defining and attaching all metadata) from a generic NumPy array.

To do this, a random array was generated and from this the GeoTIFF was built.

All GeoTIFF files with their respective metadata were checked using QGIS for better visualisation.

In the below 4-steps description of the example there are two steps where, instead of using random arrays, data (but not metadata) from two existing GeoTIFF files (one multi-band and one single band) are used. This is to verify that the metadata obtained from the function is consistent.

This example is performed in four steps:

1. defining of the function

Rasterio library is used to define a function that, from a generic NumPy array, creates a GeoTIFF file by correctly attaching metadata.

The function is parametric with some optional fields and other are required.

Some parameters are passed in by the user and are required while others already have a default value but can still take another as input.

Parameters:

- Required
 - the path of GeoTIFF output file
 - data array
 - CRS
- Optional
 - Data type (default is “Byte” but also “Float32” is allowed)
 - Compression method (default is “LZW”)
 - Block x size (must be multiples of 16 and default is 64)
 - Block y size (must be multiples of 16 and default is 64)
 - Photometric (default is “RGB” but for single band file is used “Gray”)

```
def geotiff_builder(output_path, data_array, crs_input, data_type = "Byte", compression_method = "LZW", block_x_size = 64, block_y_size = 64, pm = 'RGB')
```

2. construction of a NumPy array consisting of random values, construction of the GeoTIFF file with its metadata and visualisation on QGIS.

Building a NumPy array with random values

```
In [3]: random_array = np.random.randint(255, size=(3, 10000, 10000), dtype="uint8")
random_array
```

```
Out[3]: array([[[ 68, 127, 169, ..., 44, 25, 57],
 [ 34, 66, 42, ..., 113, 115, 11],
 [162, 118, 188, ..., 204, 154, 140],
 ...,
 [211, 7, 144, ..., 58, 114, 50],
 [ 14, 144, 217, ..., 84, 179, 157],
 [ 46, 101, 13, ..., 48, 247, 190]],

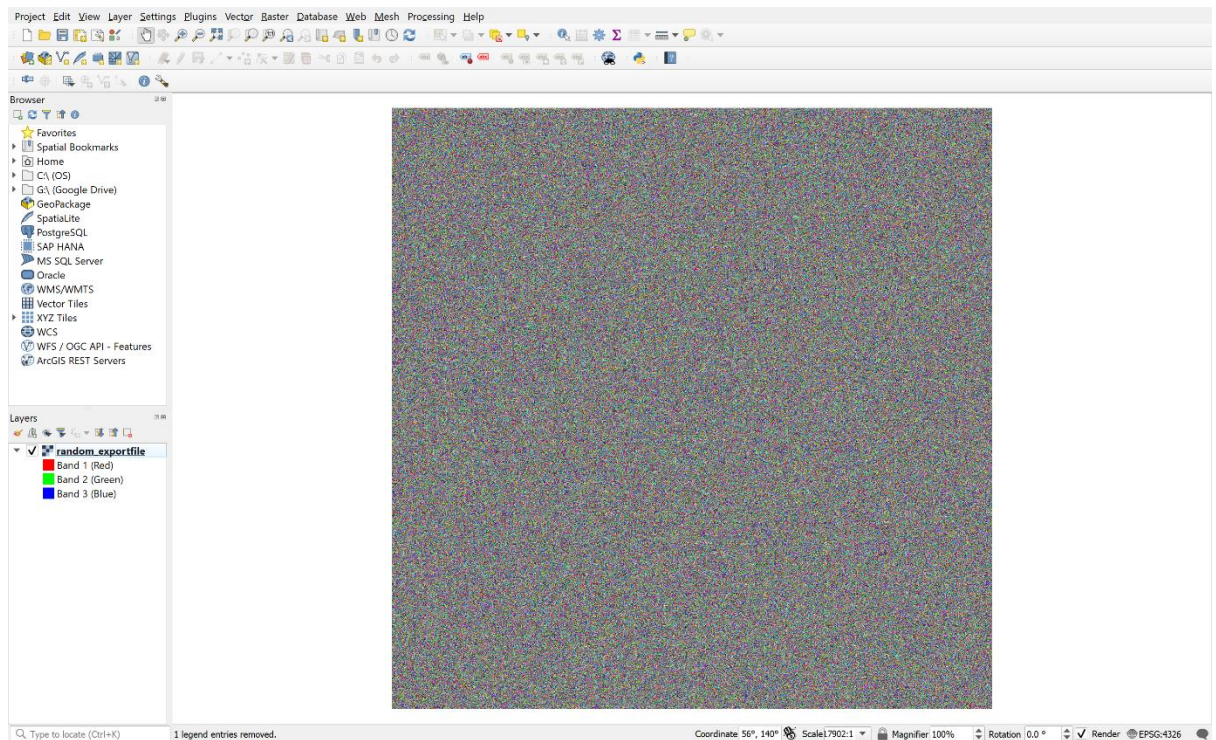
 [[180, 162, 196, ..., 198, 95, 74],
 [ 14, 70, 10, ..., 187, 103, 167],
 [215, 183, 90, ..., 120, 86, 231],
 ...,
 [152, 91, 175, ..., 51, 39, 244],
 [ 89, 191, 75, ..., 122, 204, 220],
 [105, 236, 49, ..., 88, 29, 90]],

 [[ 88, 142, 60, ..., 43, 134, 139],
 [ 95, 29, 141, ..., 22, 79, 87],
 [ 51, 30, 225, ..., 163, 69, 137],
 ...,
 [144, 85, 153, ..., 113, 78, 249],
 [194, 151, 90, ..., 237, 80, 69],
 [249, 220, 38, ..., 247, 12, 160]]], dtype=uint8)
```

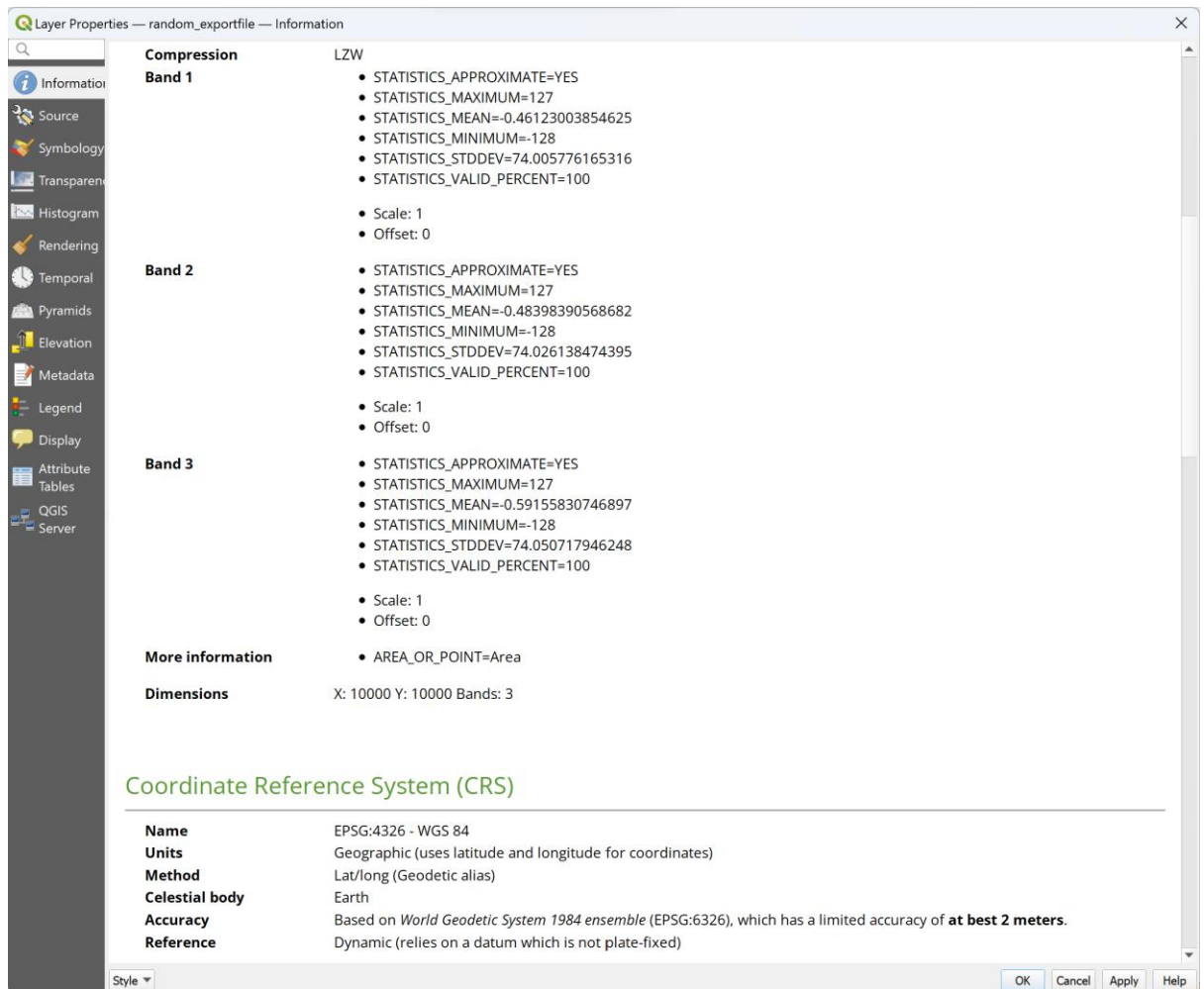
```
In [4]: print(type(random_array))
print(random_array.shape)

<class 'numpy.ndarray'>
(3, 10000, 10000)
```

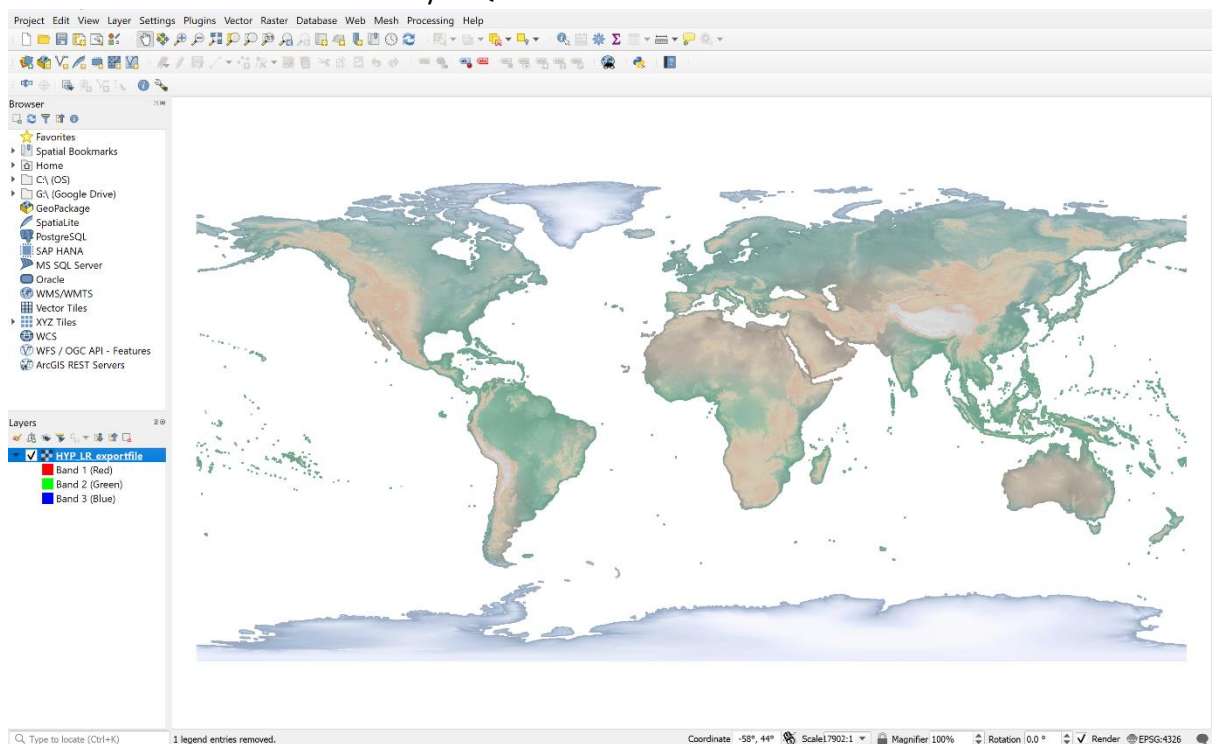
After saving the GeoTIFF file obtained from the NumPy random array to disk, it is tried to open it in QGIS to check that everything went well.

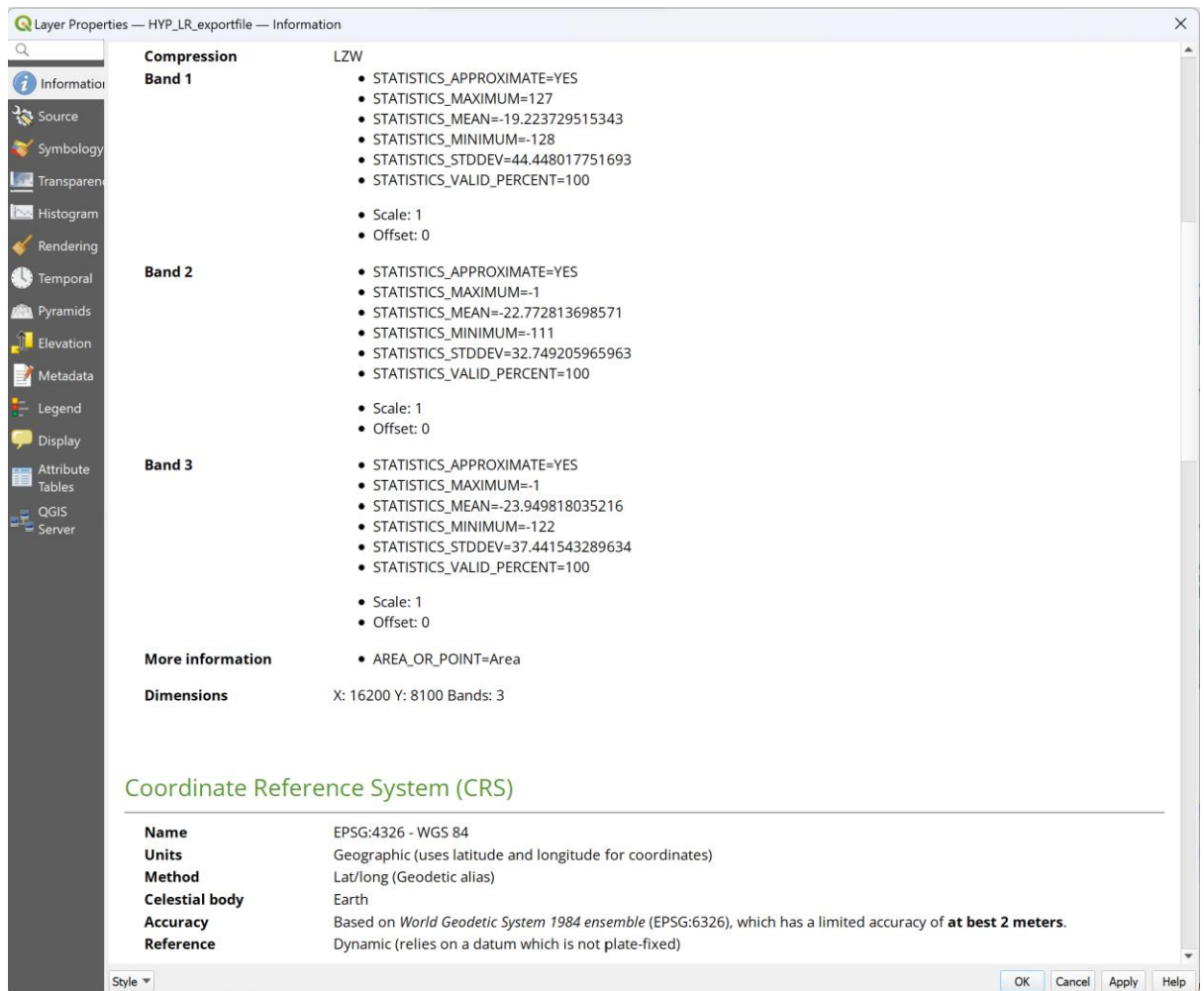


The file is loaded correctly, without returning errors and with the chosen CRS.

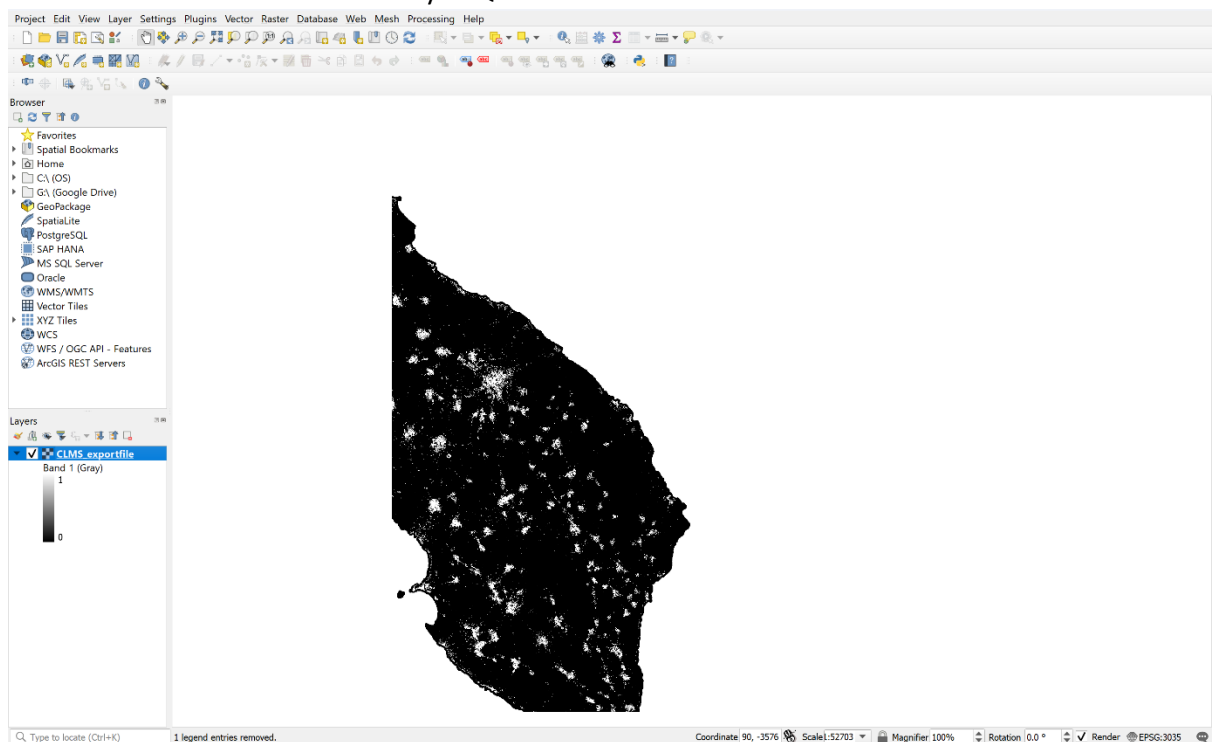


3. GeoTIFF file creation using an array of data from an existing (multi band) GeoTIFF and verification of metadata consistency in QGIS.





4. Creation of a GeoTIFF file using an array of data from an existing (single band) GeoTIFF and verification of metadata consistency in QGIS.



Layer Properties — CLMS_exportfile — Information

Information

Source

Symbology

Transparency

Histogram

Rendering

Temporal

Pyramids

Elevation

Metadata

Legend

Display

Attribute Tables

QGIS Server

Information from provider

Extent

Width

Height

Data type

GDAL Driver Description

GDAL Driver Metadata

Dataset Description

Compression

Band 1

More information

Dimensions

0.0000000000000000,-10000.0000000000000000 : 10000.0000000000000000,0.0000000000000000

10000

10000

Byte - Eight bit unsigned integer

GTiff

GeoTIFF

C:/Users/Antonio Cozzolino/OneDrive/Desktop/work/FAIRITASK/FAIRITASK-GeoTIFF_IO/numpy-geotiff/generated_GeoTIFF/CLMS_exportfile.tif

LZW

- STATISTICS_APPROXIMATE=YES
- STATISTICS_MAXIMUM=1
- STATISTICS_MEAN=-0.737502004717
- STATISTICS_MINIMUM=-1
- STATISTICS_STDDEV=0.47008641881217
- STATISTICS_VALID_PERCENT=100

- Scale: 1
- Offset: 0

- AREA_OR_POINT=Area

X: 10000 Y: 10000 Bands: 1

Coordinate Reference System (CRS)

Name

Units

Method

Celestial body

Accuracy

Reference

EPSG:3035 - ETRS89-extended / LAEA Europe

meters

Lambert Azimuthal Equal Area

Earth

Based on *European Terrestrial Reference System 1989 ensemble* (EPSG:6258), which has a limited accuracy of **at best 0.1 meters**.

Static (relies on a datum which is plate-fixed)

Identification

Style

OK

Cancel

Apply

Help