GEE Timeseries Explorer

Release 1.1

Andreas Rabe, Philippe Rufin

Sep 29, 2021

CONTENT

This documentation is structured as follows:

CHAPTER

ONE

GEE TIMESERIES EXLORER

GEE Timeseries Exlorer is a QGIS plugin for interactive exploration of temporal raster data available via the Google Earth Engine (GEE) Data Catalog.

The **GEE Timeseries Exlorer** plugin adds a panel for selecting a GEE image collection and a plot widget for visualizing temporal profiles.

1.1 Installation

In QGIS select **QGIS > Plugins > Manage and Install Plugins. . .** , search for *GEE Timeseries Exlorer* and install the plugin.

Note: this plugin relies on the *Earth Engine Python API*. Search for *Google Earth Engine* and install this plugin as well. You may also want to have a look here in case of any issue whilst installing: https://gee-community.github.io/qgis-earthengine-plugin/ In order to access Earth Engine, you must first authorize your Google account for the use of the python API. This in turn requires a Google Account with access to Earth Engine.

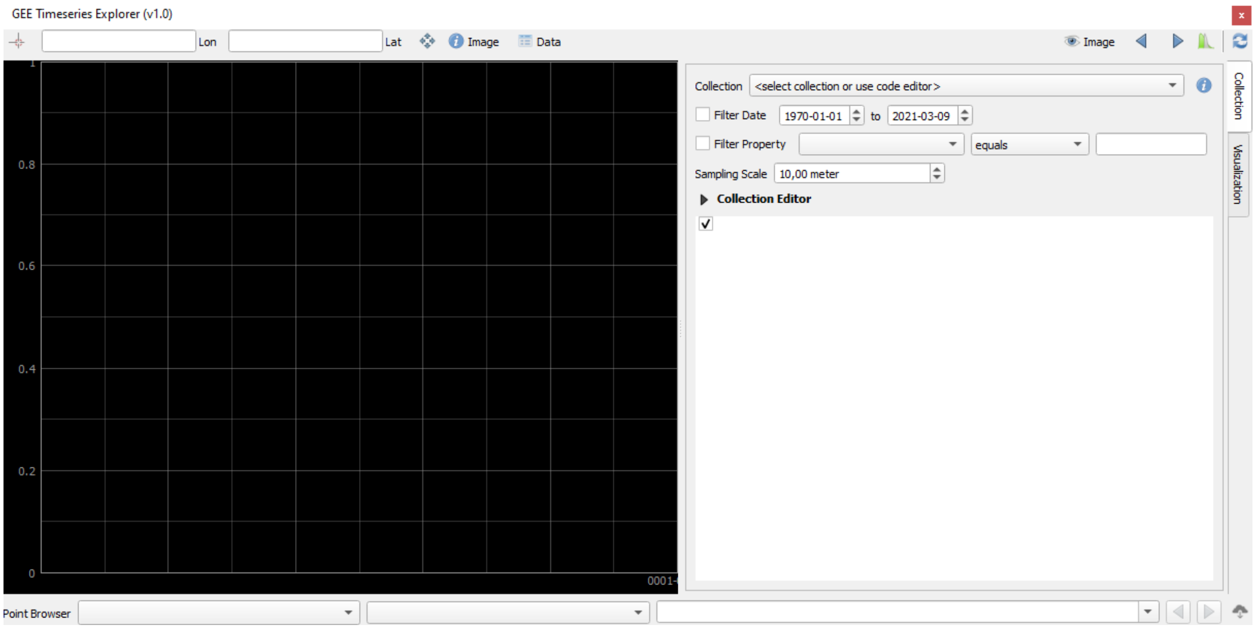
- In QGIS select **QGIS > Settings > User Profiles > Open Active Profile Folder**
- Navigate to **python > plugins > ee_plugin > extlibs_windows** (**_linux** / **_darwin** on Linux/Mac systems respectively) and copy the full path to this directory
- Open the OSGeo4W Shell / Terminal and run the following:

```
py3_env
python
import sys
sys.path.append(r"Paste path to directory here")
import ee
ee.Authenticate()
```

Follow the instructions in your browser and copy the authentication code into the OSGeo4W / Terminal.

1.2 Getting started

Overview In the toolbar click to show the **GEE Timeseries Exlorer** panel.



GEE Timeseries Explorer (v1.0)

Collection <select collection or use code editor>

Filter Date 1970-01-01 to 2021-03-09

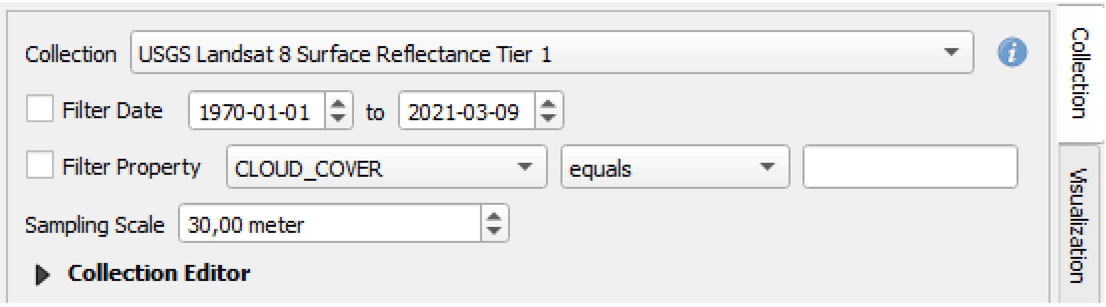
Filter Property equals

Sampling Scale 10,00 meter

There are two main panels: The **Plot Window** on the left and the **Collection** and **Visualization** settings on the right.

Load a Collection The **Collection** tab on the right panel allows you to select a predefined *Image Collection* from the list and filter it by time interval and/or metadata properties. The subsection **Collection Editor** provides the python code used to access the imagery and enables full control on the code executed by the user, having access to the entire *Earth Engine Data Catalogue* (https://developers.google.com/earth-engine/datasets).

For a quick start, the code snippet for accessing the *USGS Landsat 8 Surface Reflectance Tier 1* image collection is already prepared.



Collection USGS Landsat 8 Surface Reflectance Tier 1

Filter Date 1970-01-01 to 2021-03-09

Filter Property CLOUD_COVER equals

Sampling Scale 30,00 meter

Click on the info button next to the collection list to open and inspect the *USGS Landsat 8 Surface Reflectance Tier 1* description.

Visualize Time Series This section displays the spectral time series of the user defined collection and time frame. Click on *Activate point selection tool* in the upper left corner of the plugin and select at least one spectral band from the list. Here we use the predefined collection *Landsat TM-ETM-OLI Surface Reflectance Cloud Masked* which combines the Landsat sensors into one collection and applies masking based on quality bands (hint: check the *Collection Editor* for details). Set **Filter Date** to `1984-01-01` to `2020-12-31` to make use of the entire Landsat archive. Furthermore it can be useful to edit **Filter Property** to only consider scenes with less than 70% cloud cover.

Then, click into the map canvas to select a point location and to read the temporal profile data.

When changing the date range or filtered metadata properties, click on 🔄 Read point profile to re-read the current location.

In the top panel, click on Data to retrieve the displayed time series as raw data:

It provides access to the raw data with information on the unique image id, the geographic coordinates, the date of acquisition and the spectral values for all bands selected.

Visualize Single Images Apart from plot based time series, you can visualize entire images and image aggregates. Open the **Visualization** tab on the right panel. In the subsection **Temporal Binning** you can specify a temporal window for which the visualisation will be rendered. By default, this corresponds to a single date, but can be expanded to a date range by increasing the *Length* parameters value and/or type (Day, Month, Year). The yellow vertical line (single date) or box (multi date) in the **Plot Window** illustrates the given choice graphically and can further be used to change the temporal window interactively by clicking inside the plot window.

Secondly, select the desired image visualization in the **Band Rendering** subsection. For RGB composites use *Multiband* as *Render Type* and select an input band for each color channel, e.g by selecting `SWIR2` as red band, `NIR` as green band and `RED` as blue band. By default, the **Min / Max** values are estimated using percentiles (default: `2%` to `98%`). Furthermore, we checked *Show full map canvas extent* to visualize all scenes acquired on the specified date that fall into our current map canvas extent.

Lastly, click **Apply** to estimate the values and make sure the image visibility is toggled on 👁 Image.

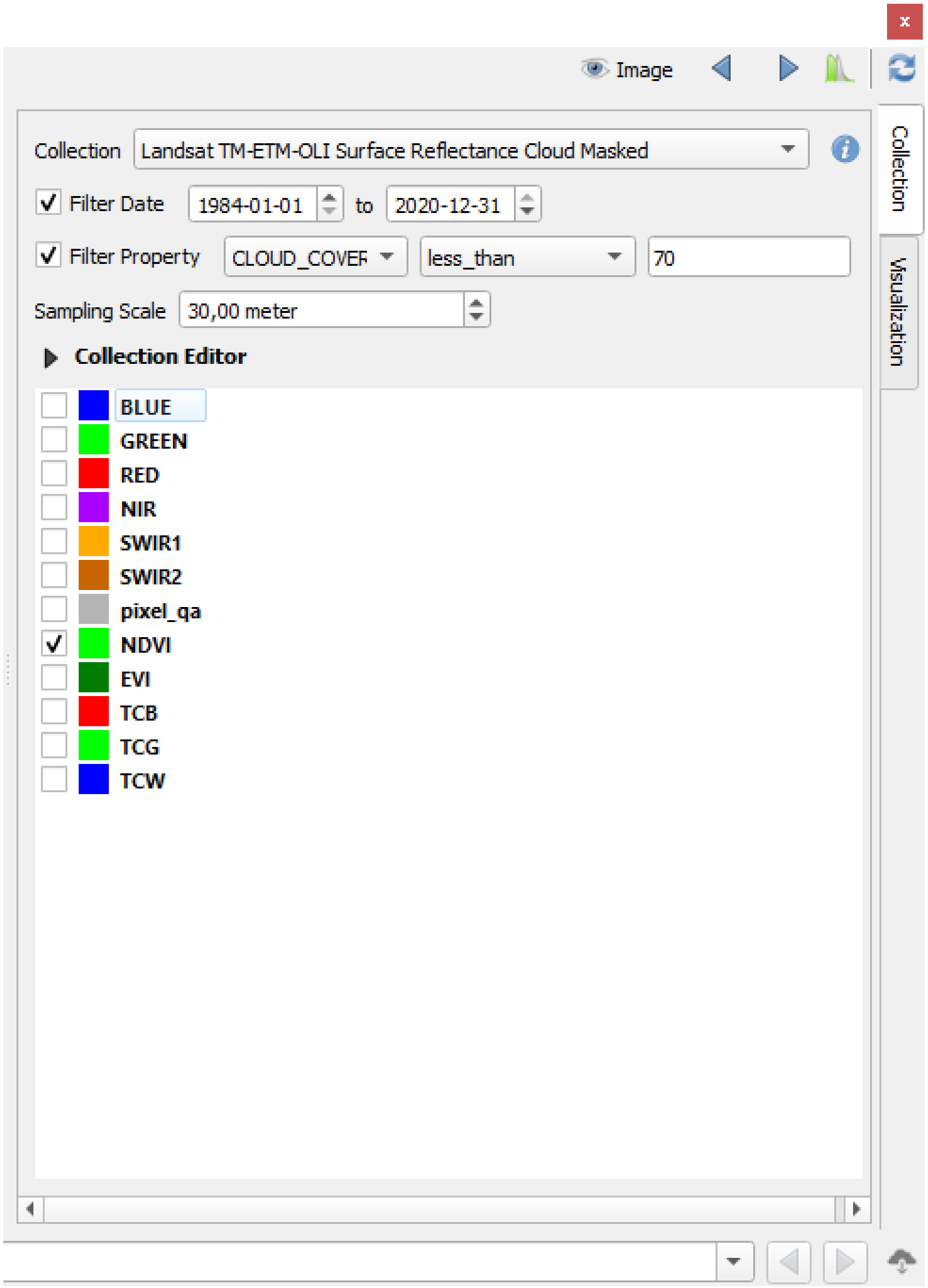
Also try to use the ◀ ▶ buttons in the upper right corner of the plugin to jump to the previous/next observation dates or time frames.

Visualize Temporally Aggregated Images In case of Landsat data it is usefull to not only visualize data at a specific date in time, but to aggregate multiple observations over a date range (e.g. the revisiting time of 16 days) and to also visualize observations from neighbouring overflight pathes at the same time. Furthermore, the aggregation of time series into pixel based statistics is usefull for preserving variance whilst reducing the dimensionality of the data. Pixel-based calculations over time are referred to as *Reducers* in Earth Engine. For each band, you can select from a variety of statistical reducers and visualize them accordingly.

For example, let's visualize the median of the selected band combination for our entire map canvas using all imagery acquired in 2018:

Download Time Series Data Often we need time series of spectral bands or indices for multiple locations in space. *GEE-TSE* allows to specify a point layer that is loaded in QGIS for which the time series of the selected bands and time frame can be downloaded to csv-files. To do so, select a point layer as input in the **Point Browser** panel at the bottom of the plugin and click ⬇ in the lower right corner. Specify a target folder in which the .csv-files for each feature of the point layer are stored.

Furthermore, you can select attributes of the point layer and use the list or the ◀ ▶ buttons to reload the plot and image visualization for the given feature.



Collection: Landsat TM-ETM-OLI Surface Reflectance Cloud Masked

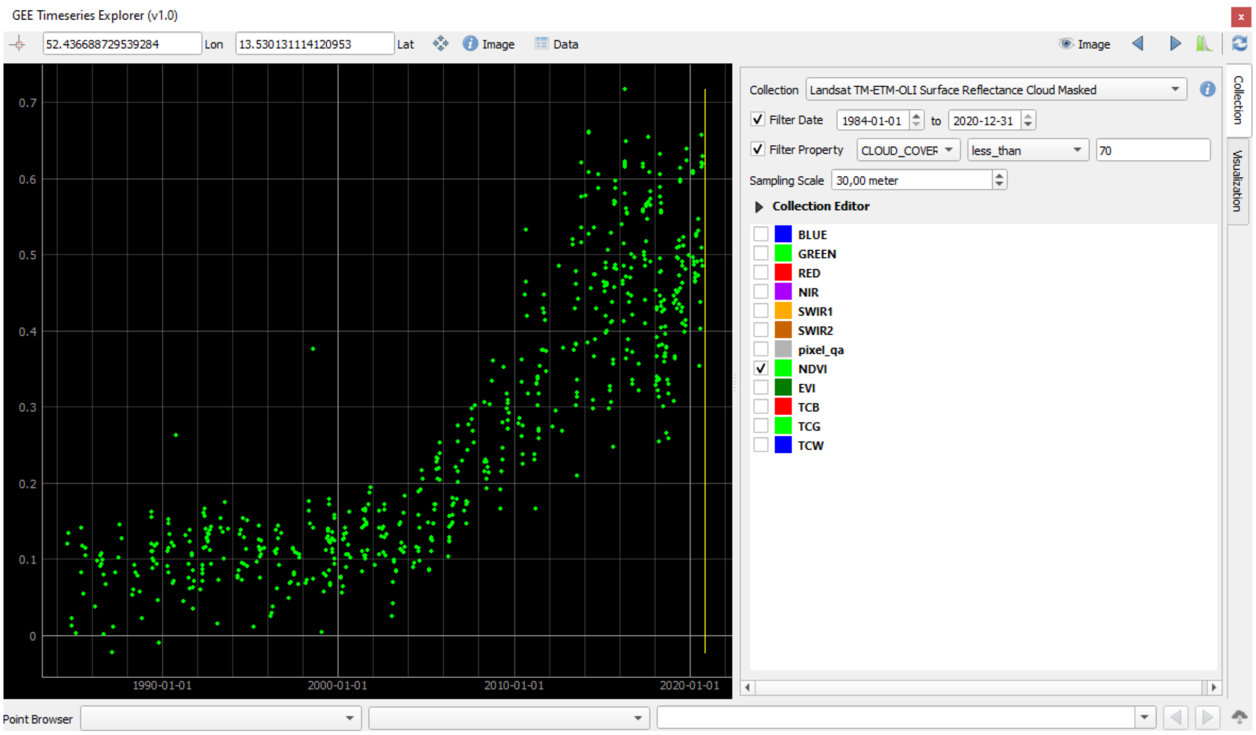
☑ Filter Date 1984-01-01 to 2020-12-31

☑ Filter Property CLOUD_COVEF less_than 70

Sampling Scale 30,00 meter

Collection Editor

- □ 🟦 BLUE
- □ 🟩 GREEN
- □ 🟥 RED
- □ 🟪 NIR
- □ 🟧 SWIR1
- □ 🟫 SWIR2
- □ ⬜ pixel_qa
- ☑ 🟩 NDVI
- □ 🟩 EVI
- □ 🟥 TCB
- □ 🟩 TCG
- □ 🟦 TCW



GEE Timeseries Explorer (v1.0)

52.436688729539284 Lon 13.530131114120953 Lat

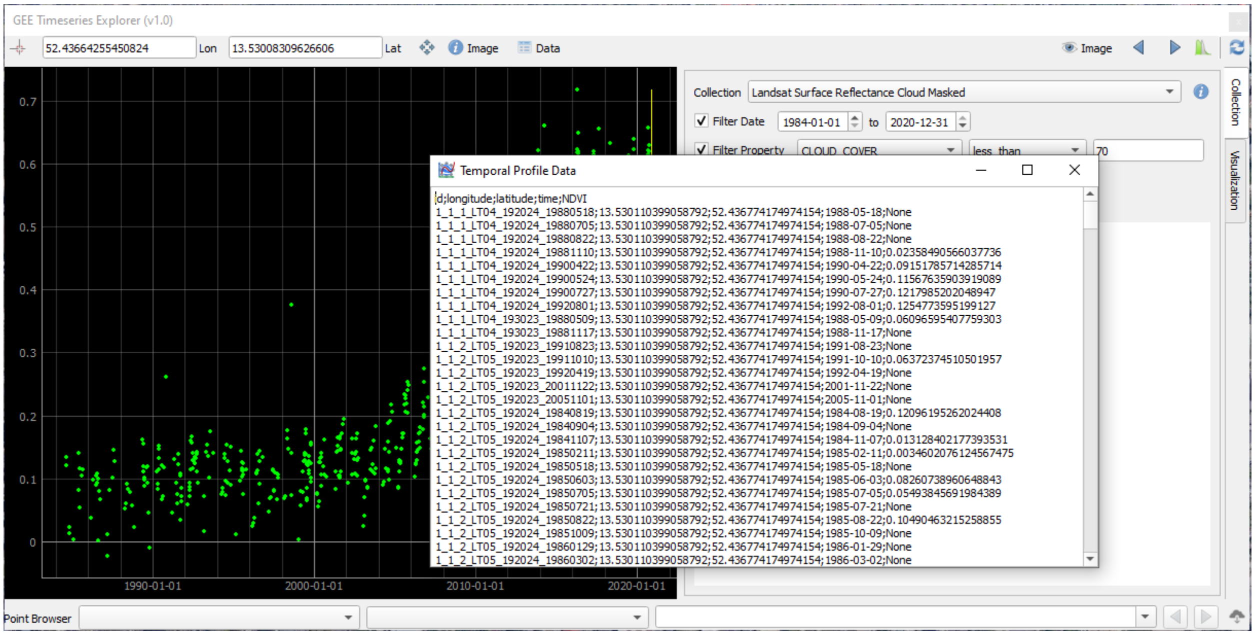
Collection Landsat TM-ETM-OLI Surface Reflectance Cloud Masked

☑ Filter Date 1984-01-01 to 2020-12-31

☑ Filter Property CLOUD_COVER less_than 70

Sampling Scale 30,00 meter

- □ 🟦 BLUE
- □ 🟩 GREEN
- □ 🟥 RED
- □ NIR
- □ SWIR1
- □ SWIR2
- □ pixel_qa
- ☑ 🟩 NDVI
- □ 🟩 EVI
- □ 🟥 TCB
- □ 🟩 TCG
- □ 🟦 TCW



GEE Timeseries Explorer (v1.0)

52.43664255450824 Lon 13.53008309626606 Lat

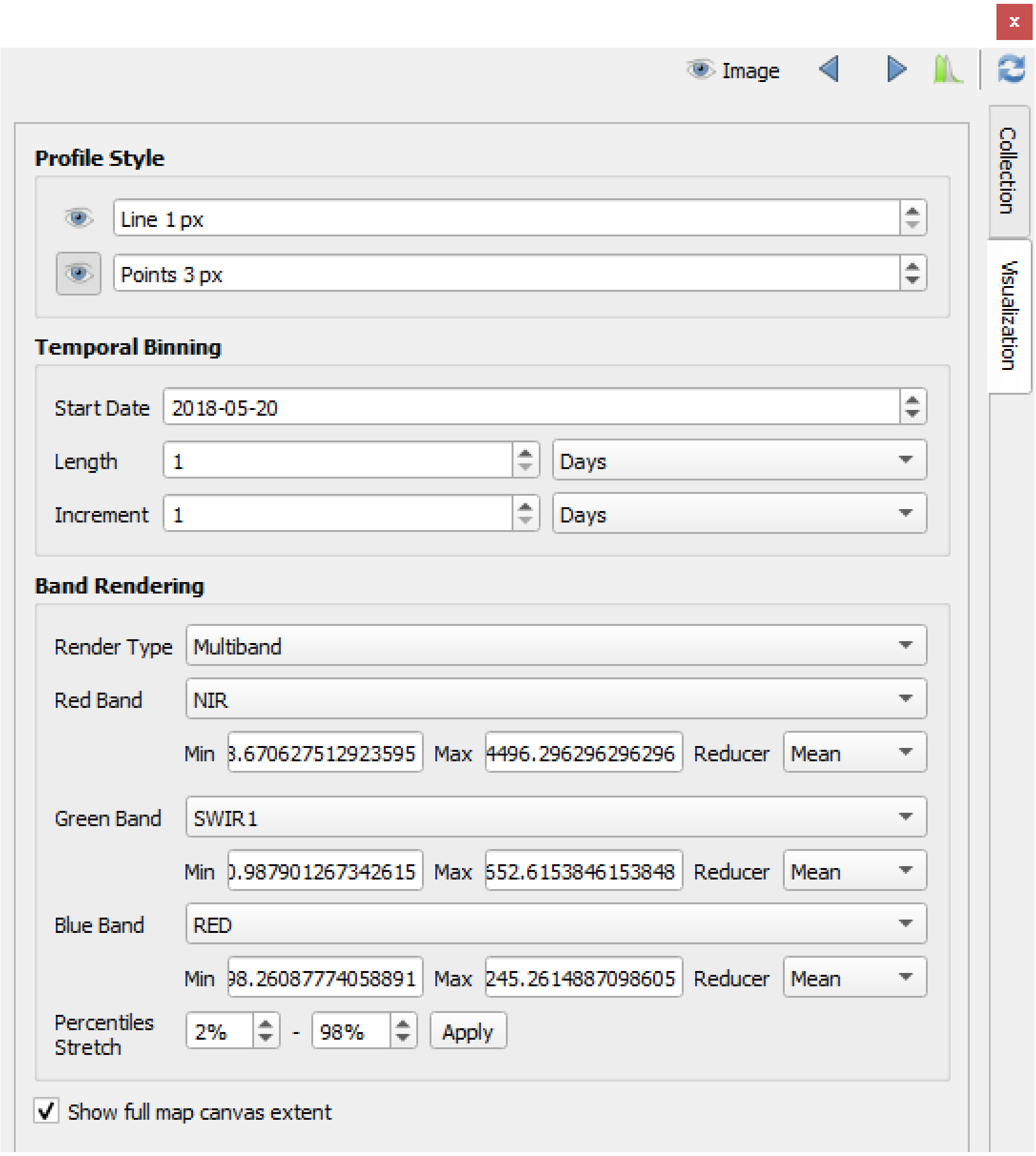
Collection Landsat Surface Reflectance Cloud Masked

☑ Filter Date 1984-01-01 to 2020-12-31

☑ Filter Property CLOUD_COVER less_than 70

Temporal Profile Data

```
id;longitude;latitude;time;NDVI
1_1_1_LT04_192024_19880518;13.530110399058792;52.436774174974154;1988-05-18;None
1_1_1_LT04_192024_19880705;13.530110399058792;52.436774174974154;1988-07-05;None
1_1_1_LT04_192024_19880822;13.530110399058792;52.436774174974154;1988-08-22;None
1_1_1_LT04_192024_19881110;13.530110399058792;52.436774174974154;1988-11-10;0.023584905660377736
1_1_1_LT04_192024_19900422;13.530110399058792;52.436774174974154;1990-04-22;0.0915178571428571 4
1_1_1_LT04_192024_19900524;13.530110399058792;52.436774174974154;1990-05-24;0.1156763590391908 9
1_1_1_LT04_192024_19900727;13.530110399058792;52.436774174974154;1990-07-27;0.121798520204894 7
1_1_1_LT04_192024_19920801;13.530110399058792;52.436774174974154;1992-08-01;0.125477359519912 7
1_1_1_LT04_193023_19880509;13.530110399058792;52.436774174974154;1988-05-09;0.0609659540775930 3
1_1_1_LT04_193023_19881117;13.530110399058792;52.436774174974154;1988-11-17;None
1_1_2_LT05_192023_19910823;13.530110399058792;52.436774174974154;1991-08-23;None
1_1_2_LT05_192023_19911010;13.530110399058792;52.436774174974154;1991-10-10;0.0637237451050195 7
1_1_2_LT05_192023_19920419;13.530110399058792;52.436774174974154;1992-04-19;None
1_1_2_LT05_192023_20011122;13.530110399058792;52.436774174974154;2001-11-22;None
1_1_2_LT05_192023_20051101;13.530110399058792;52.436774174974154;2005-11-01;None
1_1_2_LT05_192024_19840819;13.530110399058792;52.436774174974154;1984-08-19;0.1209619526202440 8
1_1_2_LT05_192024_19840904;13.530110399058792;52.436774174974154;1984-09-04;None
1_1_2_LT05_192024_19841107;13.530110399058792;52.436774174974154;1984-11-07;0.013128402177393531
1_1_2_LT05_192024_19850211;13.530110399058792;52.436774174974154;1985-02-11;0.003460207612456747 5
1_1_2_LT05_192024_19850518;13.530110399058792;52.436774174974154;1985-05-18;None
1_1_2_LT05_192024_19850603;13.530110399058792;52.436774174974154;1985-06-03;0.0826073896064884 3
1_1_2_LT05_192024_19850705;13.530110399058792;52.436774174974154;1985-07-05;0.0549384569198438 9
1_1_2_LT05_192024_19850721;13.530110399058792;52.436774174974154;1985-07-21;None
1_1_2_LT05_192024_19850822;13.530110399058792;52.436774174974154;1985-08-22;0.1049046321525885 5
1_1_2_LT05_192024_19851009;13.530110399058792;52.436774174974154;1985-10-09;None
1_1_2_LT05_192024_19860129;13.530110399058792;52.436774174974154;1986-01-29;None
1_1_2_LT05_192024_19860302;13.530110399058792;52.436774174974154;1986-03-02;None
```



Profile Style

Line 1 px

Points 3 px

Temporal Binning

Start Date 2018-05-20

Length 1 Days

Increment 1 Days

Band Rendering

Render Type Multiband

Red Band NIR

Min 3.670627512923595 Max 4496.296296296296 Reducer Mean

Green Band SWIR1

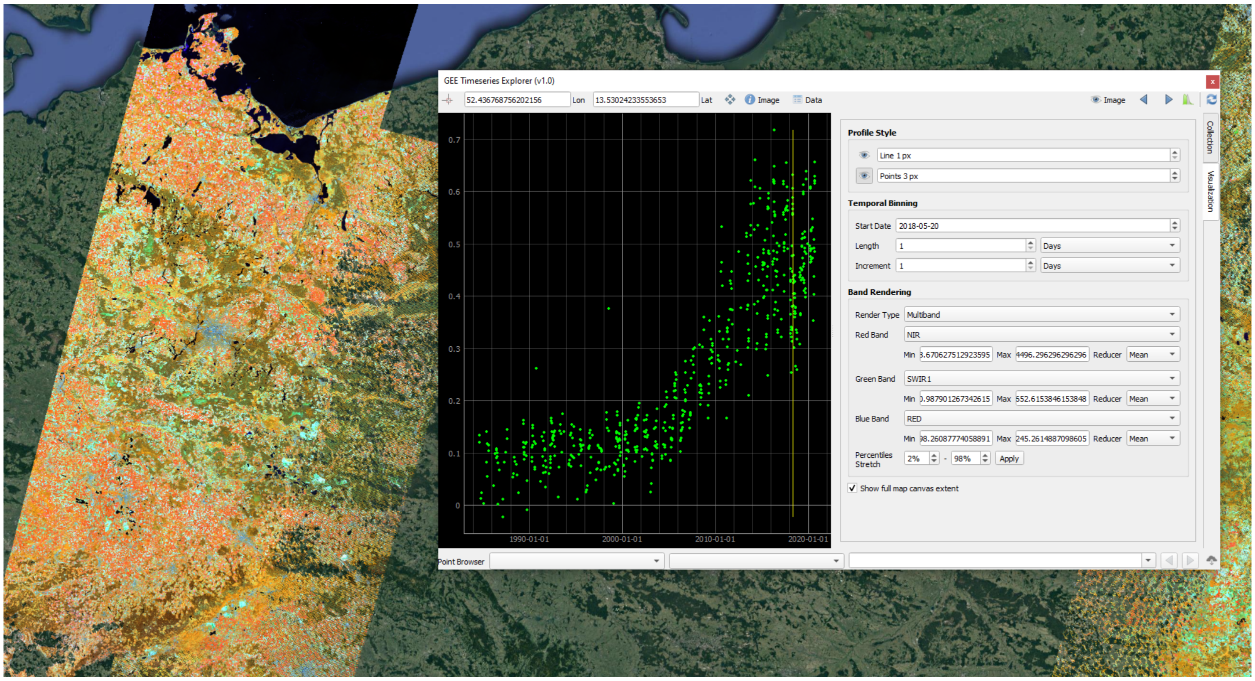
Min 0.987901267342615 Max 552.6153846153848 Reducer Mean

Blue Band RED

Min 98.26087774058891 Max 245.2614887098605 Reducer Mean

Percentiles Stretch 2% - 98% Apply

☑ Show full map canvas extent



GEE Timeseries Explorer (v1.0)

52.436768756202156 Lon 13.53024233553653 Lat

Profile Style

Line 1 px

Points 3 px

Temporal Binning

Start Date 2018-05-20

Length 1 Days

Increment 1 Days

Band Rendering

Render Type Multiband

Red Band NIR

Min 3.670627512923595 Max 4496.296296296296 Reducer Mean

Green Band SWIR1

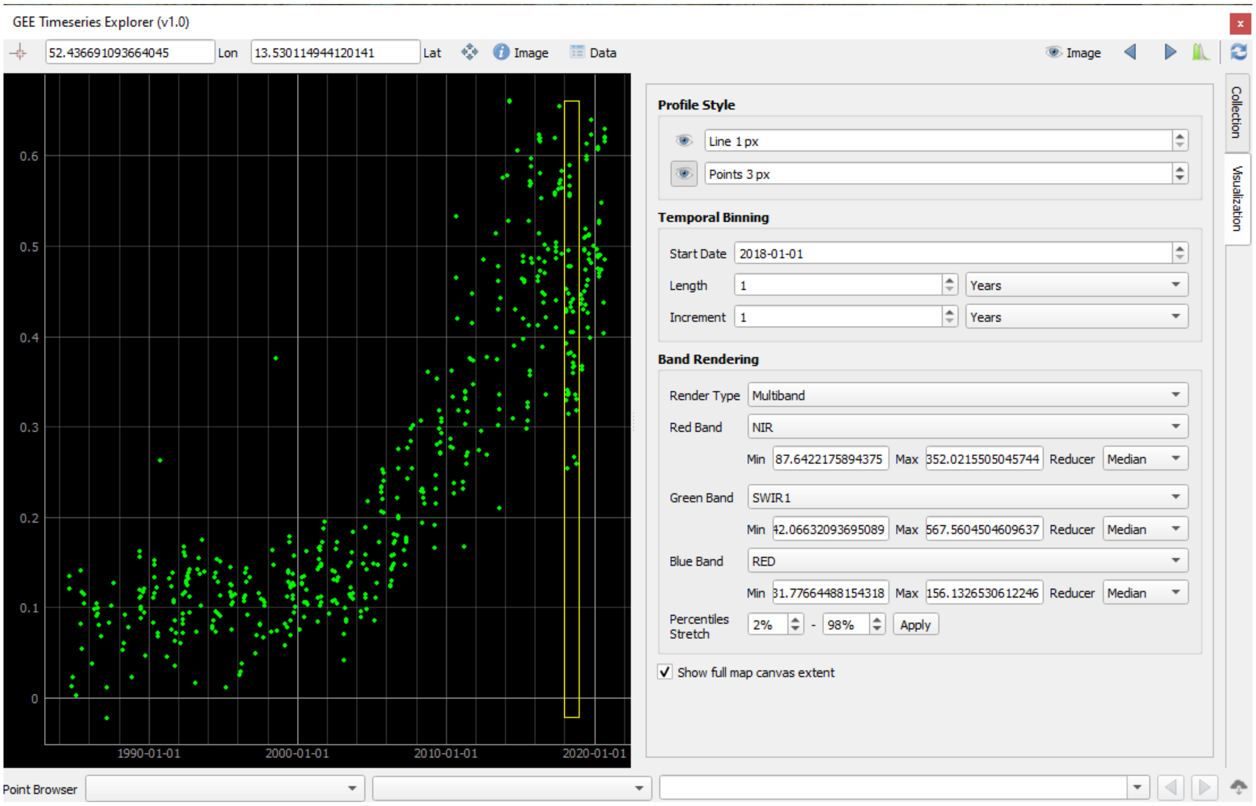
Min).987901267342615 Max 552.6153846153848 Reducer Mean

Blue Band RED

Min 98.26087774058891 Max 245.2614887098605 Reducer Mean

Percentiles Stretch 2% - 98%

☑ Show full map canvas extent



GEE Timeseries Explorer (v1.0)

52.436691093664045 Lon 13.530114944120141 Lat

Profile Style

Line 1 px

Points 3 px

Temporal Binning

Start Date 2018-01-01

Length 1 Years

Increment 1 Years

Band Rendering

Render Type Multiband

Red Band NIR

Min 87.6422175894375 Max 352.0215505045744 Reducer Median

Green Band SWIR1

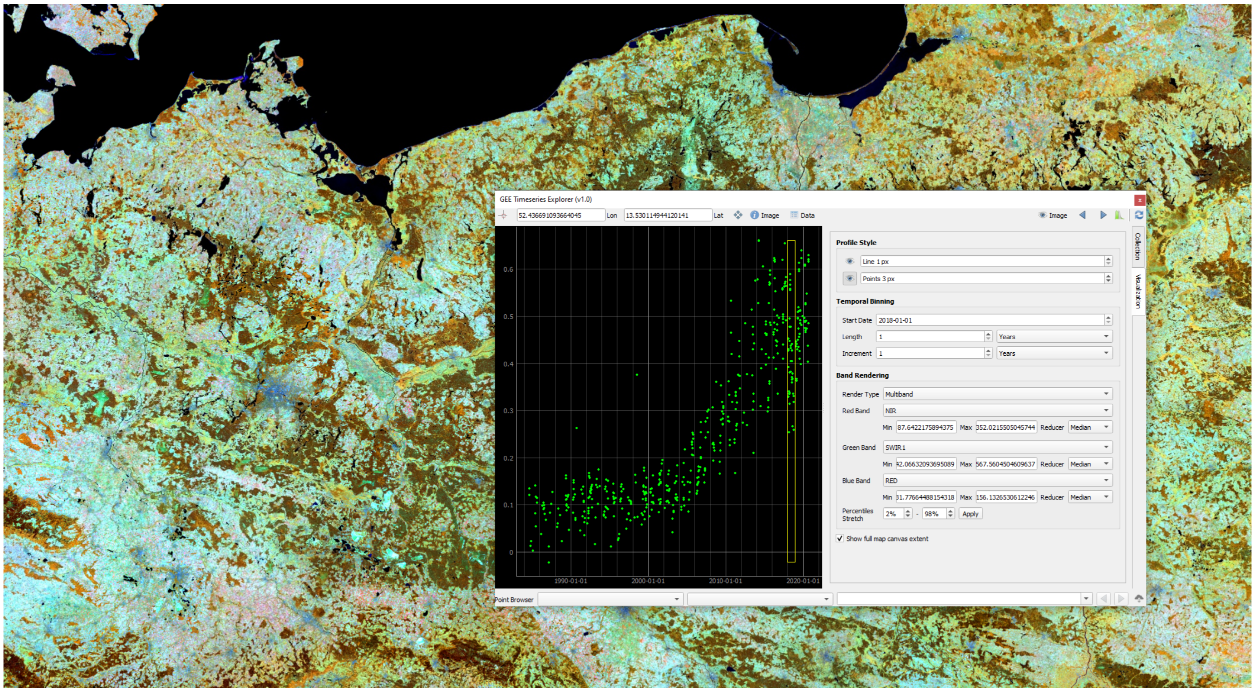
Min 42.06632093695089 Max 567.5604504609637 Reducer Median

Blue Band RED

Min 31.77664488154318 Max 156.1326530612246 Reducer Median

Percentiles Stretch 2% - 98%

☑ Show full map canvas extent


GEE Timeseries Explorer (v1.0)

52.436691093664045 Lon 13.530114944120141 Lat

Profile Style

Line 1 px

Points 3 px

Temporal Binning

Start Date 2018-01-01

Length 1 Years

Increment 1 Years

Band Rendering

Render Type Multiband

Red Band NIR

Min 87.6422175894375 Max 352.0215505045744 Reducer Median

Green Band SWIR1

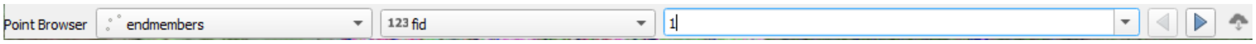
Min 42.06632093695089 Max 567.5604504609637 Reducer Median

Blue Band RED

Min 31.77664488154318 Max 156.1326530612246 Reducer Median

Percentiles Stretch 2% - 98% Apply

☑ Show full map canvas extent


Point Browser endmembers 123 fid 1

1.3 Advanced Usage

Collection Editor In the **Collection** tab, the subsection **Collection Editor** provides the python code used to access the imagery and enables full control on the code executed by the user. For instance, we might want to filter the specified collection prior to plotting and visualization.

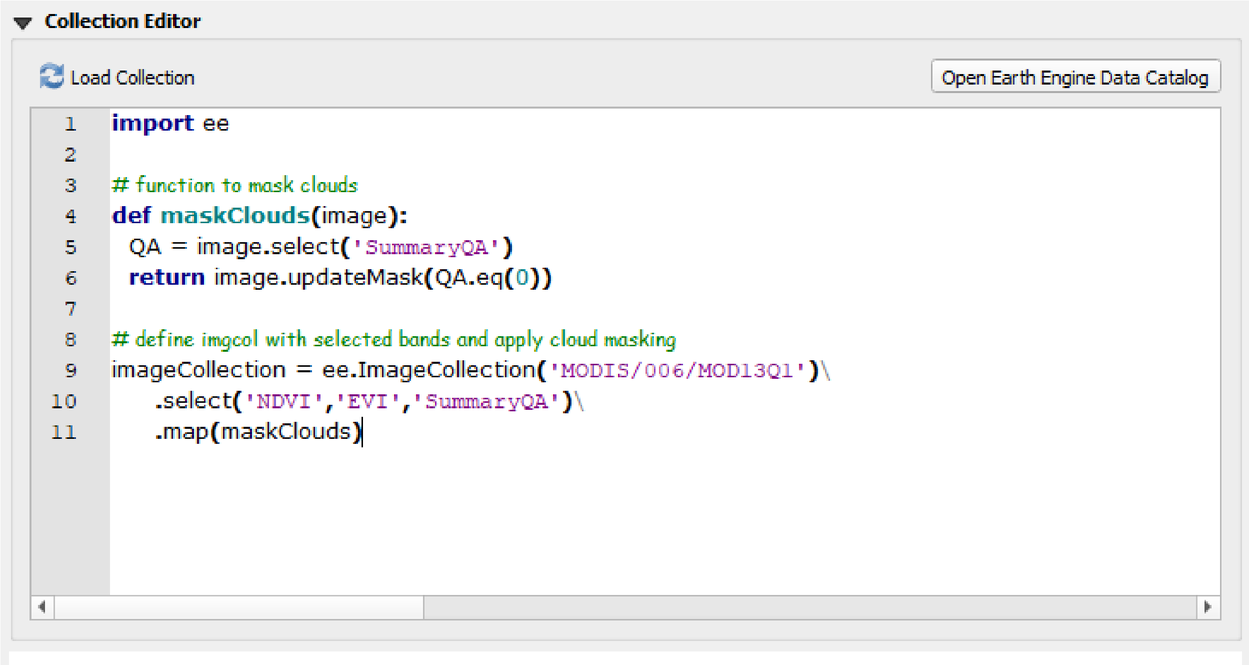
Have a look at the datasets available in the Earth Engine Catalogue by clicking on images/gee_catalogue.PNG. For now, consider the *MOD13Q1* 16-day vegetation index product (https://developers.google.com/earth-engine/datasets/catalog/MODIS_006_MOD13Q1). Open the **Collection Editor** and provide python code that loads the collection and applies a user-defined cloud masking function. **Important:** The variable name of the final ImageCollection must be *imageCollection* in order to be understood by the plugin.

```
import ee

# function to mask clouds
def maskClouds(image):
  QA = image.select('SummaryQA')
  return image.updateMask(QA.eq(0))

# define imgcol with selected bands and apply cloud masking
imageCollection = ee.ImageCollection('MODIS/006/MOD13Q1')\
     .select('NDVI','EVI','SummaryQA')\
     .map(maskClouds)
```

In the **Collection Editor** this then looks as follows:



```
import ee

# function to mask clouds
def maskClouds(image):
  QA = image.select('SummaryQA')
  return image.updateMask(QA.eq(0))

# define imgcol with selected bands and apply cloud masking
imageCollection = ee.ImageCollection('MODIS/006/MOD13Q1')\
    .select('NDVI','EVI','SummaryQA')\
    .map(maskClouds)
```

More Collections Sentinel-2

This collection returns the merged Sentinel-2A and -2B L2A (surface reflectance product) archives. It uses an aggressive cloud masking discarding images with >50% cloud cover, uses the scene classification layer to remove saturated pixels, clouds at all confidence levels, cirrus clouds, cloud shadows and snow. Furthermore, the cloud displacement index (CDI; see Frantz et al. 2018 https://doi.org/10.1016/j.rse.2018.04.046 for details) is used to improve cloud masking. The collection returns renamed bands and includes a NDVI band for convenience.

```
import ee

def maskS2scl(image):
    scl = image.select('SCL')
    sat = scl.neq(1)
    shadow = scl.neq(3)
    cloud_lo = scl.neq(7)
    cloud_md = scl.neq(8)
    cloud_hi = scl.neq(9)
    cirrus = scl.neq(10)
    snow = scl.neq(11)
    return image.updateMask(sat.eq(1)).updateMask(shadow.eq(1)).updateMask(cloud_lo.
↪eq(1)).updateMask(cloud_md.eq(1)).updateMask(cloud_hi.eq(1)).updateMask(cirrus.eq(1)).
↪updateMask(snow.eq(1)))

def maskS2cdi(image):
    cdi = ee.Algorithms.Sentinel2.CDI(image)
    return image.updateMask(cdi.gt(-0.8)).addBands(cdi)


bands = ee.List(['B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B8A', 'B11', 'B12', 'QA60',
↪'SCL', 'cdi'])
```

(continues on next page)

(continued from previous page)

```
band_names = ee.List(['blue', 'green', 'red', 'rededge1', 'rededge2', 'rededge3', 'nir',
↪'broadnir', 'swir1', 'swir2', 'QA60', 'SCL', 'CDI'])

sen = ee.ImageCollection('COPERNICUS/S2_SR')\
                .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 50))\
                .map(maskS2scl) \
                .map(maskS2cdi) \
                .select(bands, band_names)

sen = sen.map(lambda image: image.addBands(image.normalizedDifference(['broadnir', 'red
↪'])\
                                            .multiply(10000).toInt16().rename('ndvi')))

imageCollection = sen
```

1.4 Contact

Please provide feedback to Andreas Rabe (andreas.rabe@geo.hu-berlin.de) or create an issue.