Case Study 1 - Single-Image NDVI Mapping

1 Google Earth Engine registration

link

Before proceeding with the following operations, please register for a Google Earth Engine (GEE) account first. Here is a link that explains how to register a GEE account step by step.

2 Code overview

You need to download the Case Study 1.docx code file. Once you've registered for a Google Earth Engine account, launch the code editor in your browser, copy and paste the code to your own code editor, and run the code.

3 Procedures

3.1 Import L8 image collections

Code block 1:

```
// Import Landsat 8 Collection as L8
var L8_collections = ee.ImageCollection("LANDSAT/LC08/C02/T1_TOA")
    .filterDate('2017-01-01', '2017-12-31');
```

ee.ImageCollection: doc link

```
ee.ImageCollection(args)
'''
ImageCollections can be constructed from the following arguments:
  - A string: assumed to be the name of a collection,
  - A list of images, or anything that can be used to construct an image.
  - A single image.
  - A computed object - reinterpreted as a collection.

Args:
  args: ComputedObject|Image|List

Output:
  ImageCollection
'''
```

ee.ImageCollection.filterDate: doc link

```
ImageCollection.filterDate(start, end)
'''
Shortcut to filter a collection by a date range. The start and end may be Dates, numbers (interpreted as milliseconds since
     1970-01-01T00:00:00Z), or strings (such as `1996-01-01T08:00`). Based on `system:time_start`.

Args:
  start: Date|Number|String The start date (inclusive).
```

```
  end: Date|Number|String, optional. The end date (exclusive). Optional. If not specified, a 1-millisecond range starting
      at `start` is created.

Output:
  Collection
'''
```

3.2 Choose the study area

Code block 2:

```
// Set a point geometry at (-122.08412, 37.42189)
var Point = ee.Geometry.Point([-122.08412, 37.42189]);

// Show this location with a red point
Map.addLayer(Point, {color: 'red'}, 'Location Point');

// Set this point as center object
Map.centerObject(Point, 10);
```

ee.Geometry.Point: doc link

```
ee.Geometry.Point(coords, proj)
'''
Constructs an ee.Geometry describing a point.
For convenience, varargs may be used when all arguments are numbers. This allows creating EPSG:4326 points, e.g. ee.
    Geometry.Point(lng, lat).

Args:
  coords: List
  proj: Projection, optional
'''
```

Map.addLayer: doc link

```
Map.addLayer(eeObject, visParams, name, shown, opacity)
'''
Adds a given EE object to the map as a layer.
Returns the new map layer.

Args:
  eeObject: Collection|Feature|Image|RawMapId
  visParams: FeatureVisualizationParameters|ImageVisualizationParameters, optional
  name: String, optional
  shown: Boolean, optional
  opacity: Number, optional
'''
```

3.2.1 Visualization in map

You can show/hide the drawn point by clicking the 'layers' button on the top right corner of the map (Fig. 1).

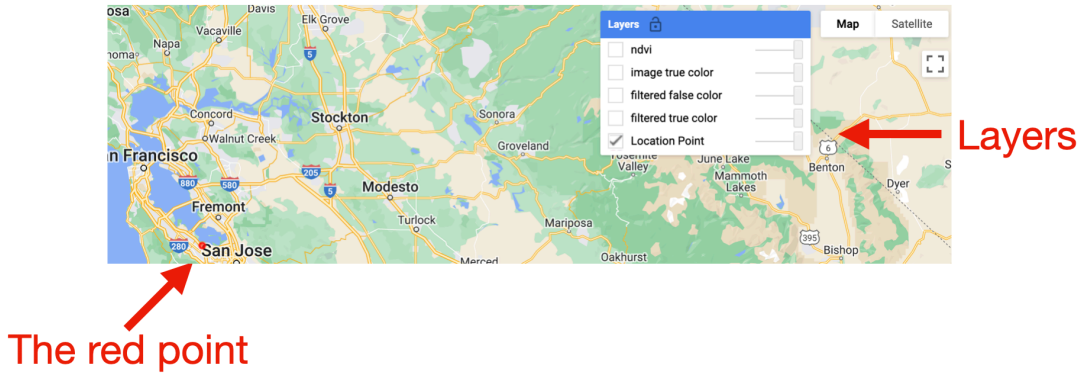

Fig. 1

3.3 Filter the images inside the study area

Code block 3:

```
// Spatial filtering
var filtered = L8_collections.filterBounds(Point)
print('filtered', filtered)
print('size', filtered.size())
```

ee.ImageCollection.filterBounds: doc link

```
ImageCollection.filterBounds(geometry)
'''
Returns the filtered collection.

Args:
  this.collection: Collection
  geometry: ComputedObject|FeatureCollection|Geometry
'''
```

3.4 Show the images in true color

Code block 4:

```
// true color
Map.addLayer(
  filtered.min(),// showing the lowest cloud coverage; to remove the cloud
  {min:0, max:0.3, bands:['B4', 'B3', 'B2']}, //visParams
  'filtered true color', // name string
  true // shown
);
```

ee.ImageCollection.min: doc link

```
ee.ImageCollection.min()
'''
Reduces an image collection by calculating the minimum value of each pixel across the stack of all matching bands. Bands
    are matched by name.

Args:
    None

Output:
    Image
'''
```

3.4.1 Visualization in map

You can show/hide the drawn map by clicking the 'layers' button on the top right corner of the map (Fig. 2).

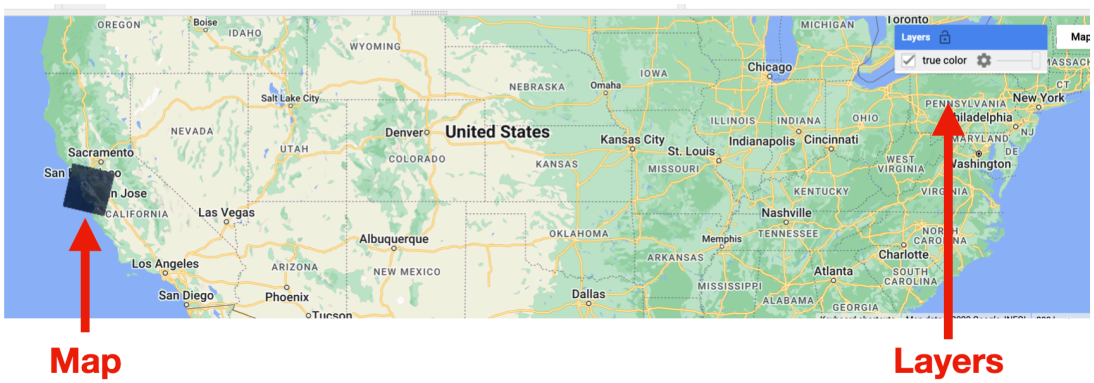

Fig. 2

3.4.2 What is true color?

Refer to the Landsat 8 documentation for a description of the bands.

What is true color (link 1, link 2)

3.5 Show the images in false color

Code block 5:

```
// false color: vegetation looks really red
Map.addLayer(
  filtered.min(),// showing the lowest cloud coverage; to remove the cloud
  {min:0, max:0.3, bands:['B5', 'B4', 'B3']},//visParams
  'filtered false color',//name string
  true //shown
);
```

3.5.1 Visualization in map

You can show/hide the drawn map by clicking the 'layers' button on the top right corner of the map (Fig. 3).

Fig. 3

3.5.2 What is false color?

What is false color (link 1, link 2)

3.6 Show the top image

Code block 6:

```
// Find the top image
var image = ee.Image(filtered.first())
// True color
Map.addLayer(
  image,
  {min:0, max:0.3, bands:['B4', 'B3', 'B2']}, //visParams
  'image true color', // name string
  true // shown
);
```

ee.Image: doc link

```
ee.Image(args)
'''
An object to represent an Earth Engine image. This constructor accepts a variety of arguments:
  - A string: an EarthEngine asset id,

  - A string and a number: an EarthEngine asset id and version,

  - A number or ee.Array: creates a constant image,

  - A list: creates an image out of each list element and combines them into a single image,

  - An ee.Image: returns the argument,

  - Nothing: results in an empty transparent image.

Args:
  args: Image|List, optional
'''
```

ee.ImageCollection.first: doc link

```
ImageCollection.first()
'''
Returns the first entry from a given collection.

Args:
  this.imagecollection: ImageCollection
'''
```

3.6.1 Visualization in map

You can show/hide the drawn map by clicking the 'layers' button on the top right corner of the map (Fig. 4).

Fig. 4

3.7 Calculate the NDVI from top image

Code block 7:

```
// Extract nir and red band from image
var nir = image.select('B5');
var red = image.select('B4');
// Calculate NDVI = (nir - red) / (nir + red)
// All clouds, shadows, and waters would have low NDVI.
// NDVI can be kind of a neat way to find data that is not contaminated by clouds.
var ndvi = (nir.subtract(red).divide(nir.add(red))); // doing pixel-wise calculation
var ndvi = image.normalizedDifference(['B5', 'B4'])
```

ee.Image.select: doc link

```
Image.select(var_args)
'''
Selects bands from an image.
Returns an image with the selected bands.

Args:
  this.image: Image
  var_args: VarArgs
'''
```

ee.Image.normalizedDifference: doc link

```
Image.normalizedDifference(bandNames)
'''
Computes the normalized difference between two bands. If the bands to use are not specified, uses the first two bands. The
    normalized difference is computed as (first - second) / (first + second). Note that the returned image band name is
    ·nd·, the input image properties are not retained in the output image, and a negative pixel value in either input band
     will cause the output pixel to be masked. To avoid masking negative input values, use ee.Image.expression() to
    compute normalized difference.


Args:
  this.input: Image
  bandNames: List, default: null
'''
```

3.8 Show the NDVI images

Code block 8:

```
//Show the ndvi
Map.addLayer(
  ndvi,
  {min:0, max:0.7}, //visParams
  'ndvi', // name string
  true // shown
);
```

3.8.1 Visualization in map

You can show/hide the drawn map by clicking the 'layers' button on the top right corner of the map (Fig. 5).

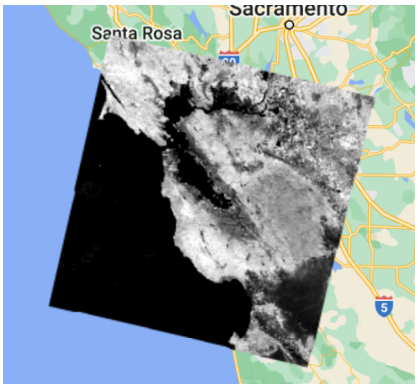

Fig. 5

4 A more comprehensive tutorial for NDVI

link

5 To do task

5.1 Display a custom point in GEE

Task: Modify the Sec. 3.2 code to display a point of your choosing on the map. You will need to replace the coordinates in the code with coordinates representing a location you want to highlight. Ensure that the new location is marked with a colorful point and centered on the map.

Hint:

You can check the "Inspector" in the top right corner of the GEE web page. Click on the map and you can see the detailed information of that location.

5.2 Customize Band Combination for Landsat 8 Imagery in GEE

Modify the Sec. 3.4 code to display the Landsat 8 imagery using a different band combination. Instead of displaying the true color image with the bands in the order ['B4', 'B3', 'B2'] (Red, Green, Blue), you are required to display the image with the bands in a custom order of your choosing. Watch the results and think about why they happen.

5.3 Calculate and Display the Enhanced Vegetation Index (EVI) Using Landsat 8 Data in GEE

Modify the Sec. 3.7 code to calculate and display the Enhanced Vegetation Index (EVI) using Landsat 8 data. The formula for EVI is:

$EVI = 2.5 \times \frac{(NIR - Red)}{(NIR + 6 \times Red - 7.5 \times Blue + 1)}$

Hint:

You may need to use the "subtract" (doc link), "multiply" (doc link), "divide" (doc link), and "add" (doc link) function.

6 Note: Latex source code of this manual

link

If future teaching assistants wish to modify this tutorial, you can open its LaTeX source code here.