Google Earth Engine

Hands on Experience

Project 6 Estimation of Water Quality

Introduction

Water Quality from Ocean Colour Remote Sensing

Water quality is a general term used to determine whether the water is fit for anthropogenic activities. Using ocean colour, we can use the optically active constituents (OACs) in the water water column to assess the quality of the water. These OACs include the concentration of suspended particulate matter, cholophyll-a and coloured dissolved organic matter.

In this session, we would try to estimate the suspended partculate matter concentration in and around the Singapore waters using ocean colour remote sensing. We will use Sentinel-2 Multi-Sprectal Imager (MSI) to compute this product.

Images showing the variability in water colour based on variable water quality.

Sentinel-2 MSI

Sentinel-2 MSI is a multispectral sensor which has 13 spectral bands ranging from 443 nm to 2200 nm.

MSI has 4 bands at 10 m spatial resolution, 6 bands at 20 m and 3 bands at 60 m spatial resolution.

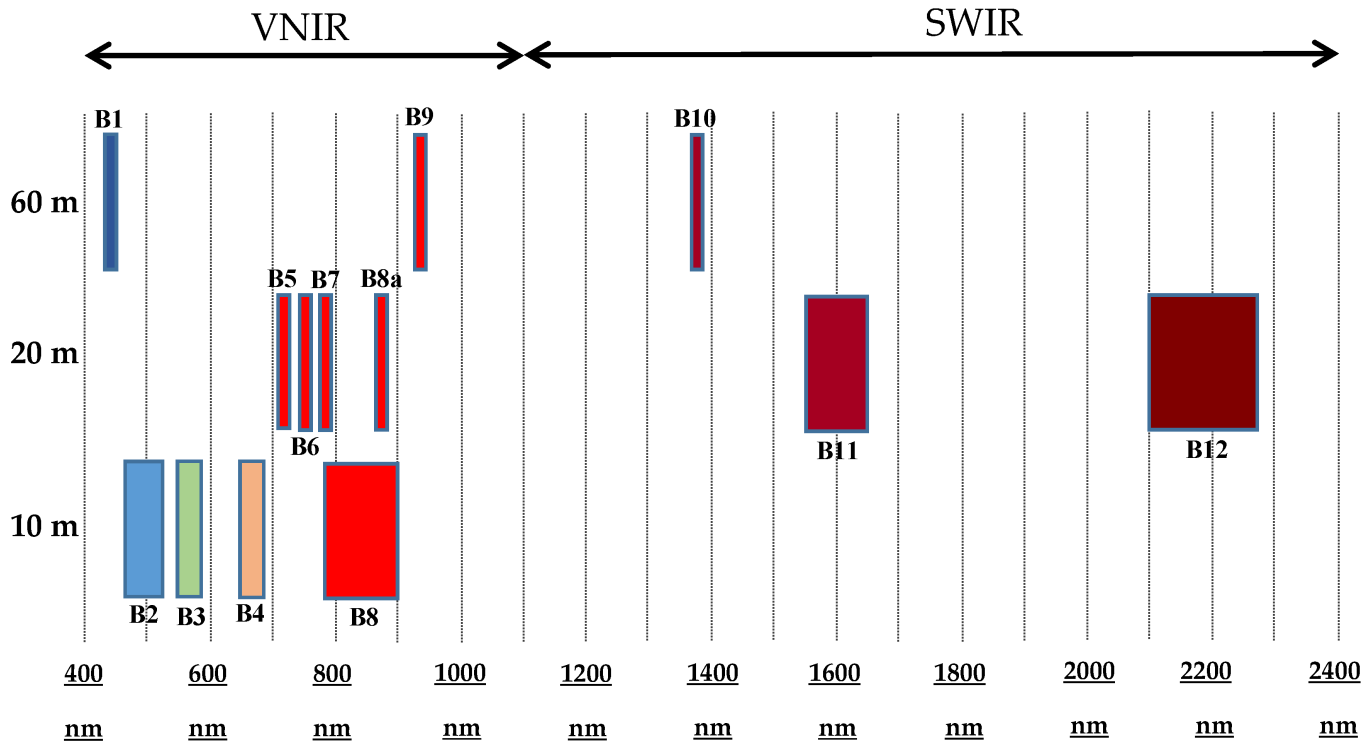
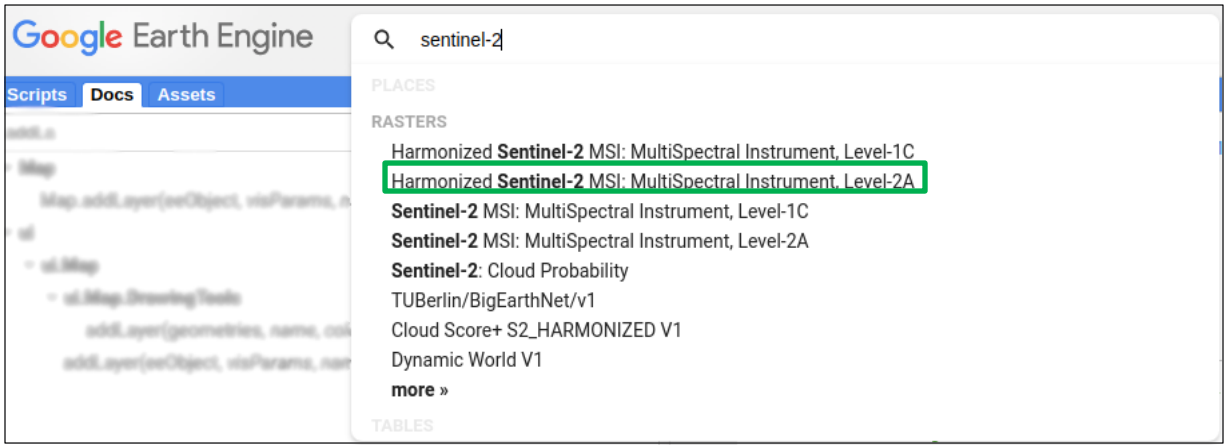


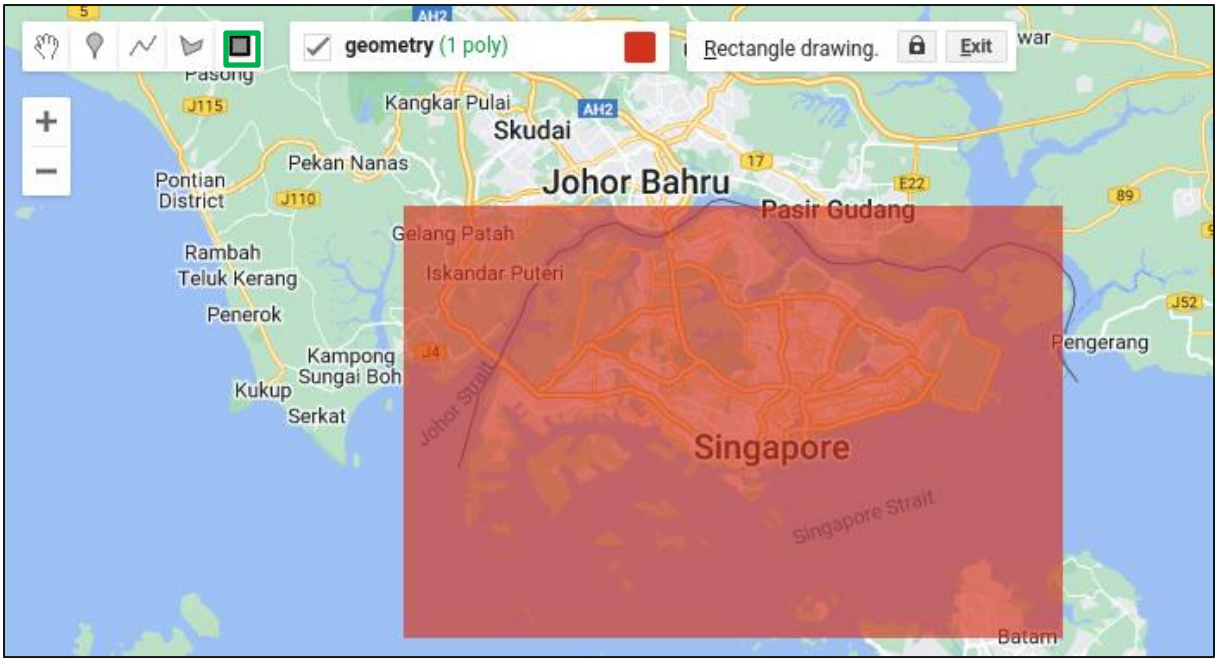
Image Source:
ESA

Guided Lab

1. Import geometry and S2 image collection

```
Imports (2 entries)
  var s2data: ImageCollection "Sentinel-2 MSI: MultiSpectral Instrument, Level-2A" (23 bands)
    type: ImageCollection
    id: COPERNICUS/S2_SR_HARMONIZED
    version: 1697423357486148
    bands: List (23 elements)
    properties: Object (21 properties)
  var geometry: Polygon, 4 vertices
    type: Polygon
    coordinates: List (1 element)
    geodesic: false
```





Guided Lab

2. Cloud Mask

```
// Import some fancy colour palettes
var palettes = require('users/gena/packages:palettes');
var palette_spm = palettes.colorbrewer.BrBG[8].reverse();

// Rename the selected geometry
var aoi = geometry;

// Write a function for masking clouds
function cloudMask(image){
  var scl = image.select('SCL');
  var mask = scl.eq(3)
              .or(scl.gte(7).and(scl.lte(10)));
  return image
      .updateMask(mask.eq(0))
      .divide(10000)
      .select("B.*");
}
```

Label	Classification
0	NO_DATA
1	SATURATED_OR_DEFECTIVE
2	CAST_SHADOWS
3	CLOUD_SHADOWS
4	VEGETATION
5	NOT_VEGETATED
6	WATER
7	UNCLASSIFIED
8	CLOUD_MEDIUM_PROBABILITY
9	CLOUD_HIGH_PROBABILITY
10	THIN_CIRRUS
11	SNOW or ICE

Guided Lab

3. Cloud and Land mask

```
function cloudLandMask(image){
   var scl = image.select('SCL');
   var mask = scl.eq(3)
          .or(scl.gte(7).and(scl.lte(10)));
   var water = scl.eq(6);
   return image
       .updateMask(mask.eq(0))
       .updateMask(water.eq(1))
       .divide(10000)
       .select("B.*");
}
```

4. Compute Suspended Sediment Concentration

```
function computeSPM(image) {
   var result = image
               .expression('(355.85 * B4 / (1-B4 / 0.1728))*0.1',
                           { 'B4': image.select('B4') }
                           );
   return result
          .rename(['spm']);
}

var spm = imageCollection
          .filterBounds(aoi)                      // Filter collection by AOI
          .filterDate('2022-01-01','2022-12-31') // Filter collection by date
          .map(cloudLandMask)                     // Apply could and land mask
          .map(computeSPM)                        // Compute SPM concentration
          .median()                               // Take median of the computed layers
          .clip(aoi);                             // Clip the output to AOI
```

Guided Lab

5. Generate true colour image as basemap

```
var rgbimg = imageCollection
             .filterBounds(aoi)
             .filterDate('2022-01-01','2022-12-31')
             .map(cloudMask)
             .median()
             .clip(aoi);
```

6. Display the generated data

```
Map.setCenter(103.805296,1.349916);    // Set Map Centre

// Add the basemap
Map.addLayer(rgbimg,
             {bands:['B4', 'B3', 'B2'], min:0.0, max:0.4},  // Visualisation parameters
              "True Colour")   // Name of the Layer



// Define Visualisation parameters for SPM data
var vis = { min: 0.0,
            max: 20,
            bands: ['spm'],
            palette: palette_spm
          };

// Add the spm data layer
Map.addLayer(spm,
             vis,
             "SPM");
```

Guided Lab

7. Show Legend on the map

```
// Creates a color bar thumbnail image for use in legend from the given colour palette.

function makeColorBarParams(palette) {
                  return {
                      bbox: [0, 0, 1, 0.1],
                      dimensions: '100x10',
                      format: 'png',
                      min: 0,
                      max: 1,
                      palette: palette
                         };
}


// Create the colour bar for the legend.

var colorBar = ui.Thumbnail(
                    { image: ee.Image.pixelLonLat().select(0),
                      params: makeColorBarParams(vis.palette),
                      style: {stretch: 'horizontal',
                              margin: '0px 8px',
                              maxHeight: '24px'}
                    }
);

// Create a panel with three numbers for the legend.

var legendLabels = ui.Panel(
         { widgets: [ ui.Label(vis.min, {margin: '4px 8px'}),
                      ui.Label( ((vis.max-vis.min) / 2+vis.min),
                                {margin: '4px 8px',
                                 textAlign: 'center',
                                 stretch: 'horizontal'}),
                      ui.Label(vis.max, {margin: '4px 8px'})
                     ],
           layout: ui.Panel.Layout.flow('horizontal') });


var legendTitle = ui.Label({ value: 'Suspended Particulate Matter (mg/L)',
                             style: {fontWeight: 'bold'} });


// Add the legendPanel to the map.

var legendPanel = ui.Panel([legendTitle, colorBar, legendLabels]);

Map.add(legendPanel);
```