

```

/*© 2022 Dominika Gulková <domee98@gmail.com>*/

/*import from assets GEE*/
var geometry =
  /* color: #d63000 */
  /* shown: false */
  /* displayProperties: [
    {
      "type": "rectangle"
    }
  ] */
  ee.Geometry.Polygon(
    [[[-58.729003248215946, -63.762950731632614],
      [-58.729003248215946, -64.45610661042129],
      [-56.927245435715946, -64.45610661042129],
      [-56.927245435715946, -63.762950731632614]]], null, false),
    table = ee.FeatureCollection("projects/ee-domcis/assets/islandboi"),
    elevation = ee.Image("users/Domcis/DEM1");

/*set sensor and bands*/
var STD_Names = ['blue', 'green', 'red', 'nir', 'swirl', 'swir2'];
var sensor = 'COPERNICUS/S2';
var sensorBands = ['B2', 'B3', 'B4', 'B8A', 'B11', 'B12'];

/*datum (yyyy-mm-dd)*/
var startDate = '2016-02-01';
var endDate = '2021-12-31';

/*mapCenter (Lon, Lat, zoom)*/
Map.setCenter(-57.85, -64.05, 10);

/*polygon study area*/
var studyArea = table;

/* minimal elevation */
var elevBounds = 0;

/*band names*/
var blue = sensorBands[0];
var green = sensorBands[1];
var red = sensorBands[2];
var nir = sensorBands[3];
var swirl = sensorBands[4];
var swir2 = sensorBands[5];

/* import sentinel2 and cloudFilter */
var satellite = ee.ImageCollection(sensor).filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30));

/* spatial filter */
var spatialFiltered = satellite.filterBounds(studyArea);

/* filter by date */
var withBands = spatialFiltered.filterDate(startDate, endDate);

/*mosaic tiles for the same date*/
function mosaicByDate(withBands){
  var imlist = withBands.toList(withBands.size());
  print(imlist);

  var unique_dates = imlist.map(function(im){
    return ee.Image(im).date().format("YYYY-MM-dd");
  }).distinct();
  print(unique_dates);
  var mosaic_imlist = unique_dates.map(function(d){
    d = ee.Date(d);
    //print(d)
    var im = withBands
      .filterDate(d, d.advance(1, "day"))
      .mosaic();
    //print(im)
    return im.set(
      "system:time_start", d.millis(),
      "system:id", d.format("YYYY-MM-dd"));
  });

  return ee.ImageCollection(mosaic_imlist);
}

var ic_m = mosaicByDate(withBands);
print(ic_m);

/*elevationData*/
var addElev = function(image) {
  return image.addBands(elevation.select('elevation'));
};
var withBands = ic_m.map(addElev);

```

```

/*****
                                Cloud Mapping
*****/

/* Cloud Mapping: Blue for identifying clouds, SWIR1 for differentiating clouds from snow */
var addSCLouds = function(image) {
  var cloudy =
  ((image.select(swirl1).divide(10000)).gte(0.12)).and((image.select(blue).divide(10000)).gte(0.2))
  .rename('Cloud_Index');
  return image.addBands(cloudy);
};

/*****
                                Saito S3 Index
*****/
/* Saito 1999 S3 Index */
var addSentS3 = function(image) {
  var snow3 = (((image.select(nir)).divide(10000))
  .multiply(((image.select(red)).divide(10000)).subtract((image.select(swirl1)).divide(10000))))
  .divide(((image.select(nir)).divide(10000)).add((image.select(red)).divide(10000))))
  .multiply(((image.select(nir)).divide(10000)).add((image.select(swirl1)).divide(10000))))
  .rename('S3index');
  return image.addBands(snow3);
};

/*****
                                Calling the appropriate Formulae
*****/

var withBands = (sensorBands == S2_Bands) ? withBands.map(addSCLouds) : withBands.map(addClouds) ;
var withBands = (sensorBands == S2_Bands) ? withBands.map(addSentS3) : withBands.map(addS3index) ;

/*****
                                NDSI
*****/
/* Normalized Difference Snow Index*/
var addNDSI = function(image) {
  var ndsi = image.normalizedDifference([green, swirl1]).rename('NDSI');
  return image.addBands(ndsi);
};
var withBands = withBands.map(addNDSI);

/*****
                                Combined Snow Index
*****/

/* If both assign snow to a pixel (and Cloud_Index does not indicate clouds) then assign snow */
var addCSI = function(image) {
  var combSnowIndex = (image.select('NDSI').gte(0.05).and(image.select('S3index').gte(0.01))
  .and(image.select('Cloud_Index').eq(0))
  .and(image.select('elevation').gte(elevBounds))
  ).rename('CSI');
  return image.addBands(combSnowIndex);
};
var withBands = withBands.map(addCSI);

/*selectCsiBand*/
var indexy = withBands.select ('CSI');
/*download whole ImageCollection*/
var batch = require('users/fitoprincipe/geetools:batch');
batch.Download.ImageCollection.toDrive(indexy, "CSI", {scale: 10});

```