//GLACIER SCALE

//EXAMPLE: BALTORO GLACIER

var ASTERemis = ee.Image("NASA/ASTER\_GED/AG100\_003"),

 ASTERVisParam = {"opacity":1,"bands":["emissivity\_band13"],"min":936,"max":991,"palette":[]},

 geometry =

 /\* color: #d63000 \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[86.76149002470473, 28.022213727262105],

 [86.76149002470473, 27.961354540750783],

 [86.83564773954848, 27.961354540750783],

 [86.83564773954848, 28.022213727262105]]], null, false),

 geometry2 =

 /\* color: #98ff00 \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[85.50556365067534, 28.278641189223965],

 [85.50556365067534, 28.207868138965274],

 [85.57869139725737, 28.207868138965274],

 [85.57869139725737, 28.278641189223965]]], null, false),

 geometry3 =

 /\* color: #d63000 \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[86.59315201272315, 28.122967399602608],

 [86.59315201272315, 27.908378532417903],

 [86.80875870217628, 27.908378532417903],

 [86.80875870217628, 28.122967399602608]]], null, false),

 geometry4 =

 /\* color: #98ff00 \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[101.83603115410801, 29.646423129346612],

 [101.83603115410801, 29.51564797975601],

 [102.09077663750645, 29.51564797975601],

 [102.09077663750645, 29.646423129346612]]], null, false),

 geometry5 =

 /\* color: #98ff00 \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[75.85639321293415, 35.965727075086285],

 [75.85639321293415, 35.497520221642546],

 [76.8808683105904, 35.497520221642546],

 [76.8808683105904, 35.965727075086285]]], null, false),

 geometry6 =

 /\* color: #0b4a8b \*/

 /\* shown: false \*/

 /\* displayProperties: [

 {

 "type": "rectangle"

 }

 ] \*/

 ee.Geometry.Polygon(

 [[[79.19785676869084, 30.862871258718318],

 [79.19785676869084, 30.683521912139373],

 [79.45603547962834, 30.683521912139373],

 [79.45603547962834, 30.862871258718318]]], null, false),

 geometry7 =

 /\* color: #bf04c2 \*/

 /\* shown: false \*/

 ee.Geometry.Polygon(

 [[[70, 45],

 [105, 45],

 [105, 25],

 [70, 25],

 [70, 45]]]),

 SouthAsiaEast = ee.FeatureCollection("users/kb621/gamdam20180404\_001\_SouthAsiaEast"),

 SouthAsiaWest = ee.FeatureCollection("users/kb621/gamdam20180404\_001\_SouthAsiaWest"),

 ERA5 = ee.ImageCollection("ECMWF/ERA5/MONTHLY"),

 SouthAsiaEast\_Scherler = ee.FeatureCollection("users/kb621/15\_rgi60\_SouthAsiaEast\_LS\_DC\_2013\_2017\_RATIO"),

 SouthAsiaWest\_Scherler = ee.FeatureCollection("users/kb621/14\_rgi60\_SouthAsiaWest\_LS\_DC\_2013\_2017\_RATIO"),

 CentralAsia\_Scherler = ee.FeatureCollection("users/kb621/13\_rgi60\_CentralAsia\_LS\_DC\_2013\_2017\_RATIO");

// 1. Filter Landsat image collection and composite into a single image

//Get Landsat 8 image collection

var l8 = ee.ImageCollection('LANDSAT/LC08/C01/T1');

var l8visParams = {bands: ['B4', 'B3', 'B2'], max: 30000};

//Filter Landsat 8 image collection by space and time

var l8spatialFiltered = l8.filterBounds(geometry5); // change geometry depending on glacier

var l8temporalFiltered = l8spatialFiltered

.filter(ee.Filter.calendarRange(2013,2015,'year')) // change year range depending on glacier

.filter(ee.Filter.calendarRange(5,10,'month'));

//Filter to find the images with less than 20% cloud cover

var lesscloudy = l8temporalFiltered

.filterMetadata('CLOUD\_COVER', 'less\_than', 20);

print(lesscloudy, 'less cloudy');

//Set visual parameters

var l8visParams\_mean = {bands: ['B4\_mean', 'B3\_mean', 'B2\_mean'], max: 30000};

//Temporally reduce the image collection by taking the mean of the

//middle 60% of pixels of the images with less than 20% cloud cover

var mean\_lesscloudy = lesscloudy

.reduce(ee.Reducer.intervalMean(20, 80));

Map.addLayer(mean\_lesscloudy, l8visParams\_mean, 'Mean of middle 60%');

// 2. Calculate the land surface temperature from the composite thermal band imagery

// Convert the raw data (composite image) to radiance

var radiance\_mean = ((mean\_lesscloudy.multiply(0.0003342)).add(0.1));

// Convert the raw data to top-of-atmosphere reflectance

var one = radiance\_mean.divide(radiance\_mean);

var k1constant = one.multiply(774.8853);

var k2constant = one.multiply(1321.0789);

var toa\_mean = k2constant.divide(((k1constant.divide(radiance\_mean)).add(1)).log());

//Calculation of surface temperature using the single channel

//atmospheric correction algorithm

var ASTERemis13 = ASTERemis.select(['emissivity\_band13']);

var ASTERemis14 = ASTERemis.select(['emissivity\_band14']);

var ASTERemis\_mean = (ASTERemis13.add(ASTERemis14)).divide(2);

var logASTERemis = (ASTERemis\_mean.multiply(0.001)).log();

var ASTERemis\_mean\_final = ASTERemis\_mean.multiply(0.001);

var lamda = (toa\_mean.multiply(toa\_mean)).divide(radiance\_mean.multiply(1324));

var delta = toa\_mean.subtract((toa\_mean.multiply(toa\_mean)).divide(1324));

var atmos\_trans = one.multiply(0.85); // change atmospheric correction depending on glacier

var LW\_up = one.multiply(1.07); // change atmospheric correction depending on glacier

var LW\_down = one.multiply(1.81); // change atmospheric correction depending on glacier

var tau1 = one.divide(atmos\_trans);

var tau2 = (LW\_down.multiply(-1)).subtract(LW\_up.divide(atmos\_trans));

var tau3 = LW\_down;

var LST\_mean2 = (((((tau1.multiply(radiance\_mean)).add(tau2)).multiply(one.divide(0.94))).add(tau3)).multiply(lamda)).add(delta);

//Map.addLayer(LST\_mean2, {bands: ['B10\_mean'], min: 250, max: 320, palette: ['5EDAFF', 'FBFF5E', 'FF8989']}, 'LST of mean image 2');

var LST\_mean2\_C = LST\_mean2.subtract(273.15);

Map.addLayer(LST\_mean2\_C, {bands: ['B10\_mean'], min: -10, max: 30, palette: ['5EDAFF', 'FBFF5E', 'FF8989']}, 'LST of mean image 2 degrees C');

// 3. Normalise land surface temperature

//Filter ERA5 Reanalysis data

var ERA5temporalFiltered =

ERA5.filter(ee.Filter.calendarRange(2013,2016,'year'))

.filter(ee.Filter.calendarRange(5,10,'month'));

//Temporally reduce the image collection by taking the mean of the middle 60% of pixels

var mean\_ERA5filtered = ERA5temporalFiltered

.reduce(ee.Reducer.intervalMean(20, 80));

var ERA5visParams\_mean = {bands: ['mean\_2m\_air\_temperature\_mean'],

min:260,max: 320};

//Set visual parameters

var visParams = {bands: ['mean\_2m\_air\_temperature'], min:260,max: 310};

//Map.addLayer(mean\_ERA5filtered, ERA5visParams\_mean, 'Mean of ERA5');

//Find mean air temperature of HMA region

var meanDictionary = mean\_ERA5filtered.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: geometry7,

scale: 30000,

maxPixels: 1e9

});

print('Mean:', meanDictionary.get('mean\_2m\_air\_temperature\_mean'));

//Produce a constant array of the mean air temperature

var constant = ee.Number(meanDictionary.get('mean\_2m\_air\_temperature\_mean'));

var constant\_array = (mean\_ERA5filtered.divide(mean\_ERA5filtered)).multiply(constant);

//Divide constant array by ERA5 composite image

var multiplier = constant\_array.divide(mean\_ERA5filtered);

//Multiply surface temperature image by multiplier image tonormalise

var LST\_mean\_norm\_C = LST\_mean2\_C.multiply(multiplier.select( "mean\_2m\_air\_temperature\_mean"));

Map.addLayer(LST\_mean\_norm\_C, {bands: ['B10\_mean'], min: -10, max: 30, palette: ['5EDAFF', 'FBFF5E', 'FF8989']}, 'LST of mean image normalised degrees C');

//Clip the calculated LST to the extent of the geometry

var dt\_c = LST\_mean\_norm\_C.divide((LST\_mean\_norm\_C.multiply(-0.065)).add(2.086)); // change relationship applied depending on glacier

var dt\_c\_clippedW = (dt\_c.select('B10\_mean')).clipToCollection(SouthAsiaWest\_Scherler);

var dt\_c\_clippedE = (dt\_c.select('B10\_mean')).clipToCollection(SouthAsiaEast\_Scherler);

var dt\_c\_clippedC = (dt\_c.select('B10\_mean')).clipToCollection(CentralAsia\_Scherler);

var dt\_c\_clipped = ee.ImageCollection([dt\_c\_clippedW, dt\_c\_clippedE, dt\_c\_clippedC]).mosaic();

//var dt\_c\_clipped = dt\_c.clipToCollection(SouthAsiaEast\_Scherler);

Map.addLayer(dt\_c\_clipped, {bands: ['B10\_mean'], min: 0, max: 200, palette: ['FF0000', '00FF00', '0000FF']}, 'dt of LST\_mean normalised degrees C clipped\_rational');