// MODIS Gap-Filled ET Processing Script for Multiple Years and Seasons

// Processes MODIS ET data for 2017-2022 across Kharif, Rabi, and Zaid seasons

// Downloads separate rasters for each season of each year

var block\_name = "Boipariguda";

// Define seasons with their date ranges

var seasons = {

'kharif': {

'start\_month': 7, // July

'start\_day': 1,

'end\_month': 10, // October

'end\_day': 31

},

'rabi': {

'start\_month': 11, // November

'start\_day': 1,

'end\_month': 2, // February (next year)

'end\_day': 28 // Will be adjusted for leap years

},

'zaid': {

'start\_month': 3, // March

'start\_day': 1,

'end\_month': 6, // June

'end\_day': 30

}

};

// Define years to process

var years = [2017, 2018, 2019, 2020, 2021, 2022];

// Load the MODIS Terra Net Evapotranspiration Gap-Filled collection

var modisET = ee.ImageCollection('MODIS/061/MOD16A2GF');

// Load your specific AOI

var geometry = ee.FeatureCollection("projects/ee-aatif/assets/MWS/" + block\_name + "\_MWS");

// Function to check if a year is a leap year

var isLeapYear = function(year) {

return ee.Number(year).mod(4).eq(0).and(

ee.Number(year).mod(100).neq(0).or(

ee.Number(year).mod(400).eq(0)

)

);

};

// Function to get date string in YYYY-MM-DD format

var getDateString = function(year, month, day) {

var monthStr = ee.Number(month).format('%02d');

var dayStr = ee.Number(day).format('%02d');

return ee.Number(year).format().cat('-').cat(monthStr).cat('-').cat(dayStr);

};

// Function to apply scale factor to ET band

var applyScaleFactor = function(image) {

var etScaled = image.select('ET').multiply(0.1);

return etScaled.copyProperties(image, ['system:time\_start']);

};

// Function to process ET for a specific season and year

var processSeasonET = function(year, seasonName, seasonInfo) {

var startDate, endDate;

if (seasonName === 'rabi') {

// Rabi season spans across two years (Nov to Feb)

var startMonth = seasonInfo.start\_month;

var startDay = seasonInfo.start\_day;

var endYear = year + 1;

var endMonth = seasonInfo.end\_month;

var endDay = seasonInfo.end\_day;

// Adjust for leap year (February has 29 days)

endDay = ee.Algorithms.If(

isLeapYear(endYear).and(ee.Number(endMonth).eq(2)),

29,

endDay

);

startDate = year + '-' + (startMonth < 10 ? '0' : '') + startMonth + '-' +

(startDay < 10 ? '0' : '') + startDay;

endDate = endYear + '-' + (endMonth < 10 ? '0' : '') + endMonth + '-' +

(ee.Number(endDay).getInfo() < 10 ? '0' : '') + ee.Number(endDay).getInfo();

} else {

// Kharif and Zaid seasons are within the same year

var startMonth = seasonInfo.start\_month;

var startDay = seasonInfo.start\_day;

var endMonth = seasonInfo.end\_month;

var endDay = seasonInfo.end\_day;

startDate = year + '-' + (startMonth < 10 ? '0' : '') + startMonth + '-' +

(startDay < 10 ? '0' : '') + startDay;

endDate = year + '-' + (endMonth < 10 ? '0' : '') + endMonth + '-' +

(endDay < 10 ? '0' : '') + endDay;

}

print('Processing:', seasonName, year, 'from', startDate, 'to', endDate);

// Filter collection by date range

var etCollection = modisET

.filterDate(startDate, endDate)

.select('ET');

// Apply scale factor

var etScaled = etCollection.map(applyScaleFactor);

// Calculate temporal sum

var sum = etScaled.sum();

// Set properties

return sum

.set('start\_date', startDate)

.set('end\_date', endDate)

.set('season', seasonName)

.set('year', year)

.set('description', 'MODIS ET ' + seasonName + ' season ' + year + ' (' + startDate + ' to ' + endDate + ')')

.set('units', 'kg/m²')

.set('scale\_factor\_applied', 0.1);

};

// Visualization parameters

var visParams = {

min: 0,

max: 500,

palette: ['white', 'blue', 'green', 'yellow', 'orange', 'red']

};

// Center map on AOI

Map.centerObject(geometry, 10);

// Process each year and season combination

for (var i = 0; i < years.length; i++) {

var year = years[i];

var seasonNames = Object.keys(seasons);

for (var j = 0; j < seasonNames.length; j++) {

var seasonName = seasonNames[j];

var seasonInfo = seasons[seasonName];

// Process ET for this season and year

var etSum = processSeasonET(year, seasonName, seasonInfo);

// Clip to AOI

var etSumClipped = etSum.clip(geometry);

// Add to map (only show 2022 for visualization)

if (year === 2022) {

Map.addLayer(etSumClipped, visParams,

'MODIS ET ' + seasonName + ' ' + year, false);

}

// Export to Earth Engine Asset

var assetDescription = block\_name + '\_MODIS\_ET\_' + seasonName + '\_' + year;

var assetId = 'projects/ee-aatif/assets/ET\_MODIS/' + block\_name + '\_MODIS\_ET\_' + seasonName + '\_' + year;

Export.image.toAsset({

image: etSumClipped,

description: assetDescription,

assetId: assetId,

scale: 500,

region: geometry,

maxPixels: 1e9,

crs: 'EPSG:4326'

});

// Calculate and print statistics

var stats = etSumClipped.reduceRegion({

reducer: ee.Reducer.mean().combine({

reducer2: ee.Reducer.minMax(),

sharedInputs: true

}),

geometry: geometry,

scale: 500,

maxPixels: 1e9

});

print('Statistics for ' + seasonName + ' ' + year + ':', stats);

}

}

print('All export tasks created. Check the Tasks tab to run the exports.');

print('Total tasks created:', years.length \* Object.keys(seasons).length);

// Create a summary for verification

// print('Processing Summary:');

// for (var i = 0; i < years.length; i++) {

// var year = years[i];

// var seasonNames = Object.keys(seasons);

// for (var j = 0; j < seasonNames.length; j++) {

// var seasonName = seasonNames[j];

// var seasonInfo = seasons[seasonName];

// var startDate, endDate;

// if (seasonName === 'rabi') {

// startDate = year + '-11-01';

// endDate = (year + 1) + '-02-28';

// } else if (seasonName === 'kharif') {

// startDate = year + '-07-01';

// endDate = year + '-10-31';

// } else { // zaid

// startDate = year + '-03-01';

// endDate = year + '-06-30';

// }

// print(year + ' ' + seasonName + ': ' + startDate + ' to ' + endDate);

// }

// }