RICHMOND

URICHMOND ABSAT
Amazon Borderlands
Spatial Analysis Team

Tutorial para mapear distúrbios florestais no sudoeste da Amazônia usando CODED, LandTrendr e MTDD

SERVIR AMAZONIA

Reygadas Langarica, Yunuen

UNIVERSITY OF RICHMOND

Índice

1. Histórico ....Error! Bookmark not defined.

2. Definições ....Error! Bookmark not defined.

3. Algoritmos 2

3.1. CODED 2

3.1.1. Mapa de distúrbios ....Error! Bookmark not defined.

3.1.2. Mapa de degradação e desmatamento ....Error! Bookmark not defined.

3.2. LandTrendr 9

3.2.1. *Mapa de distúrbios* ....Error! Bookmark not defined.

3.3. MTDD 13

3.3.1. Amostras de treinamento ....Error! Bookmark not defined.

3.3.2. Mapa de degradação e desmatamento 18

Referências ....Error! Bookmark not defined.

Tutorial para mapear distúrbios florestais no sudoeste da Amazônia usando CODED, LandTrendr e MTDD

1. Histórico

Devido à necessidade urgente de compreender os efeitos dos distúrbios florestais sobre os serviços dos ecossistemas, vários algoritmos de sensoriamento remoto focados na detecção de mudanças na vegetação foram desenvolvidos na última década. O desafio agora está em compreender qual algoritmo melhor se adapta à área de estudo do usuário e ao objetivo da pesquisa. Este tutorial compartilha nosso trabalho do Google Earth Engine (GEE) derivado da avaliação do desempenho de três algoritmos - Detecção de Degradação Contínua (CODED), Detecção de Tendências de Distúrbios e Recuperação (LandTrendr) baseada em Landsat, e Detecção de Perturbações de Tempo Multi-variável (MTDD) - para detectar e caracterizar perturbações florestais no sudoeste da Amazônia (SWA). Para todo o estudo, consulte Reygadas et al. (2021) (atualmente em revisão).

Na seção 3.3, compartilhamos códigos GEE baseados no MTDD para mapear distúrbios no SWA (isto é, Ucayali, Peru e Acre, Brasil). Tutoriais GEE completos para CODED e LandTrendr estão disponíveis gratuitamente online; neste tutorial, compartilhamos a configuração-parametrização específica para o SWA.

2. Definições

Embora seja difícil adotar um conjunto de definições que se harmonizem com a lógica fundamental por trás de cada algoritmo, nos referimos ao desmatamento, à degradação florestal e aos distúrbios florestais da seguinte forma:

- **Desmatamento.** Conversão de longo prazo ou permanente de terras florestadas em terras não florestadas (FAO, 2001).

- **Degradação das florestas.** Embora não haja uma definição amplamente aceita, geralmente é considerado um processo de longo prazo que não leva a uma mudança na cobertura da terra, mas afeta negativamente a estrutura e função da floresta (Sasaki e Putz, 2009; Schoene et al., 2007).

- **Distúrbios florestais**. Fatores que impulsionam o estado e função da floresta que podem variar de eventos de alto impacto, como incêndios ou desmatamento, a processos sutis e graduais, como os causados por secas prolongadas, insetos ou doenças (Cohen et al., 2017; McDowell et al., 2015).

Assim, consideramos tanto o desmatamento (ou seja, evento de alto impacto) quanto a degradação florestal (ou seja, processo sutil e gradual) tipos de distúrbios florestais.

3. Algoritmos

Os três algoritmos usam valores de reflexão de superfície do Landsat Thematic Mapper (TM), Landsat Enhanced Thematic Mapper+ (ETM+) e Landsat Operational Land Imager (OLI).

3.1. CODED

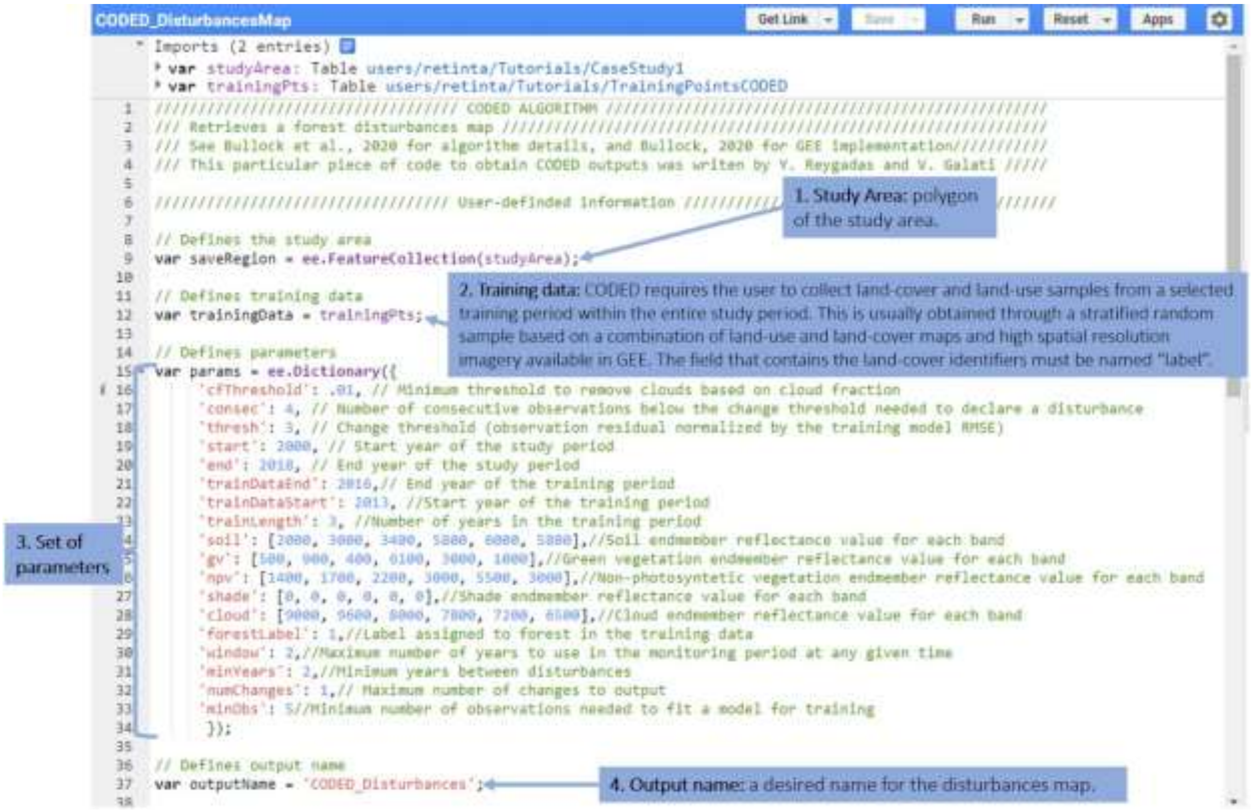
O CODED v0 utiliza séries temporais de frações de membros e NDFI para detectar e caracterizar distúrbios que podem mais tarde ser classificados como degradação ou desmatamento. Veja Bullock et al. (2020) para detalhes do algoritmo e Bullock (2020) para a implementação do GEE.

3.1.1. Mapa de distúrbios

Esta seção demonstra como executar este código para produzir um mapa de distúrbios florestais.

Insumos

O usuário tem que inserir as seguintes entradas:



Executando o Código

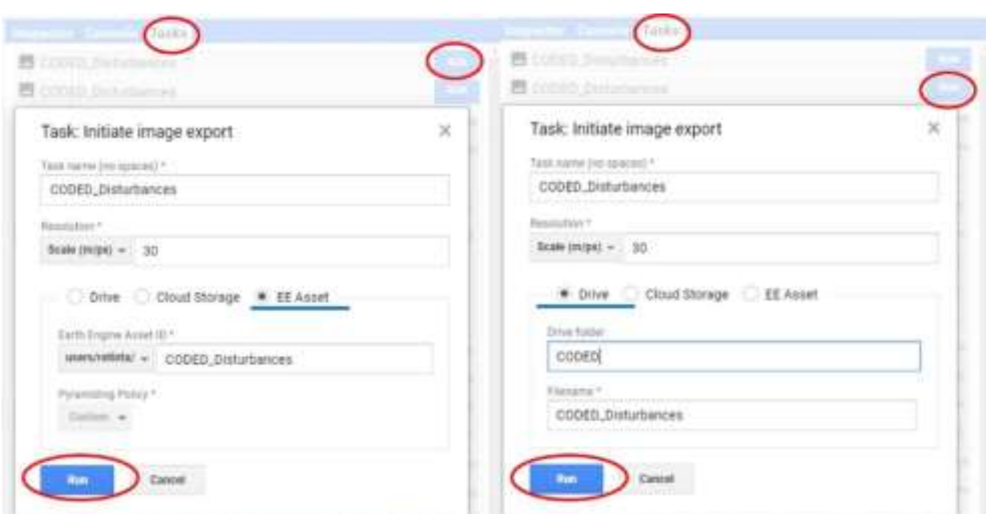
Execute o código clicando em run. Uma vez terminado, vá até o console de tarefas e clique em run para exportar os resultados como um ativo GEE e como um raster.

CODED_DisturbancesMap

```
/////////////////////////////Gets CODED main results/////////////////////////////////////

// Adds CODED utility functions
var codedUtils = require('users/retinta/Tutorials:CODED_changeDetection');
var dataUtils = require('users/retinta/Tutorials:CODED_dataUtils');

// Runs CODED
var results = codedUtils.submitCODED(saveRegion, params, trainingData);

// Turns array columns into images
var disturbances = dataUtils.makeImage(results, 0, 'dist_', params.get('start'), params.get('end'));
var magnitude = dataUtils.makeImage(results, 1, 'mag_', params.get('start'), params.get('end'));
var postChange = dataUtils.makeImage(results, 2, 'post_', params.get('start'), params.get('end'));
var difference = dataUtils.makeImage(results, 3, 'dif_', params.get('start'), params.get('end'));
var forestFlag = dataUtils.makeImage(results, 4, 'forest_', params.get('start'), params.get('end'));

var disturbanceBands = disturbances.addBands([magnitude, postChange, difference]);

var save_output = ee.Image(dataUtils.reduceBands(ee.Image(disturbanceBands), params)
                          .addBands(forestFlag.select(0)) // Forest flag for first year
                          .setMulti(params));

//////////////////////////////// Exports the results ///////////////////////////////////////

//Exports results as a GEE asset
Export.image.toAsset({
  image: save_output,
  description: outputName,
  assetId: outputName,
  maxPixels: 1e13,
  scale: 30,
  region: saveRegion,
  pyramidingPolicy: {
  '.default': 'mode'
  }
});

//Exports results as a raster to drive
Export.image.toDrive({
  image: save_output,
  description: outputName,
  region: saveRegion,
  scale: 30
});

//////////////////////////////////////////////////////////////////////////////////////////
print("All done");
```

Detects disturbances and estimates their characteristics

Exports the results as a GEE asset (used as input of the degradation/deforestation map) and as a raster (can be visualized in any GIS)



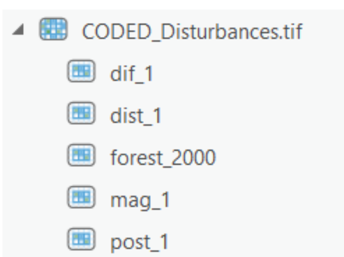
Output

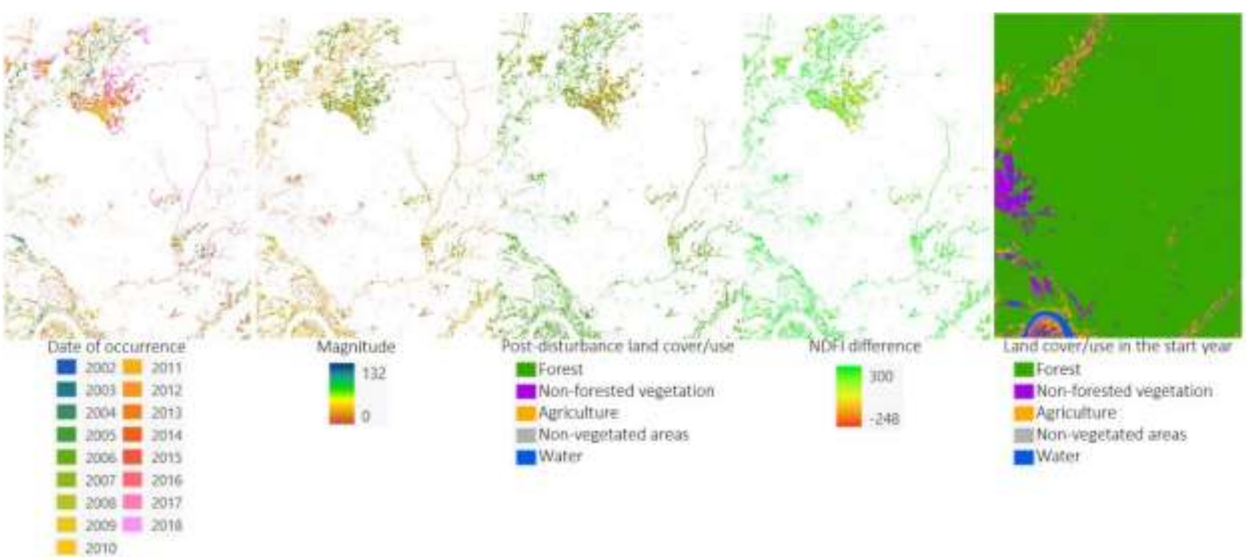
Este código produz um mapa de perturbações com quatro camadas por perturbação (isto é, data de ocorrência, magnitude da mudança, cobertura da terra após a perturbação, diferença NDFI de antes e depois da perturbação) e uma última camada indicando o tipo de cobertura da terra no ano inicial.

GEE asset

Index	Name	Type	Dimensions	CRS	Nominal Scale	Min	Max
0	dist_1	signed int16	981x1287 px	EPSG:4326	30.000000000000004	2002	2018
1	mag_1	signed int16	981x1287 px	EPSG:4326	30.000000000000004	3	132
2	post_1	signed int16	981x1287 px	EPSG:4326	30.000000000000004	0	5
3	dif_1	signed int16	981x1287 px	EPSG:4326	30.000000000000004	-248	300
4	forest_2000	signed int16	981x1287 px	EPSG:4326	30.000000000000004	1	5

Raster (".tif" file)



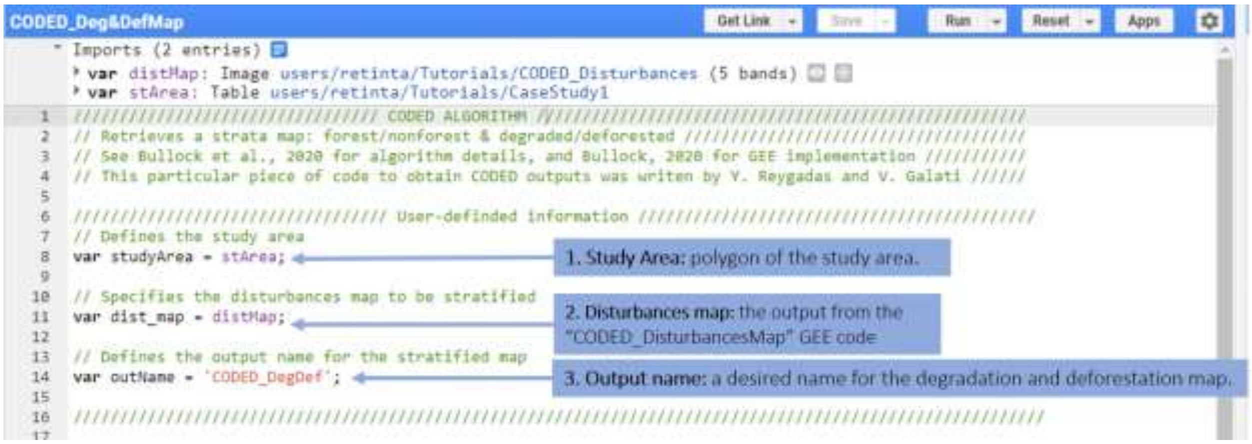


3.1.2. Mapa de degradação e desmatamento

Esta seção demonstra como executar este código para produzir um mapa classificado em floresta, nãofloresta, degradação e desmatamento.

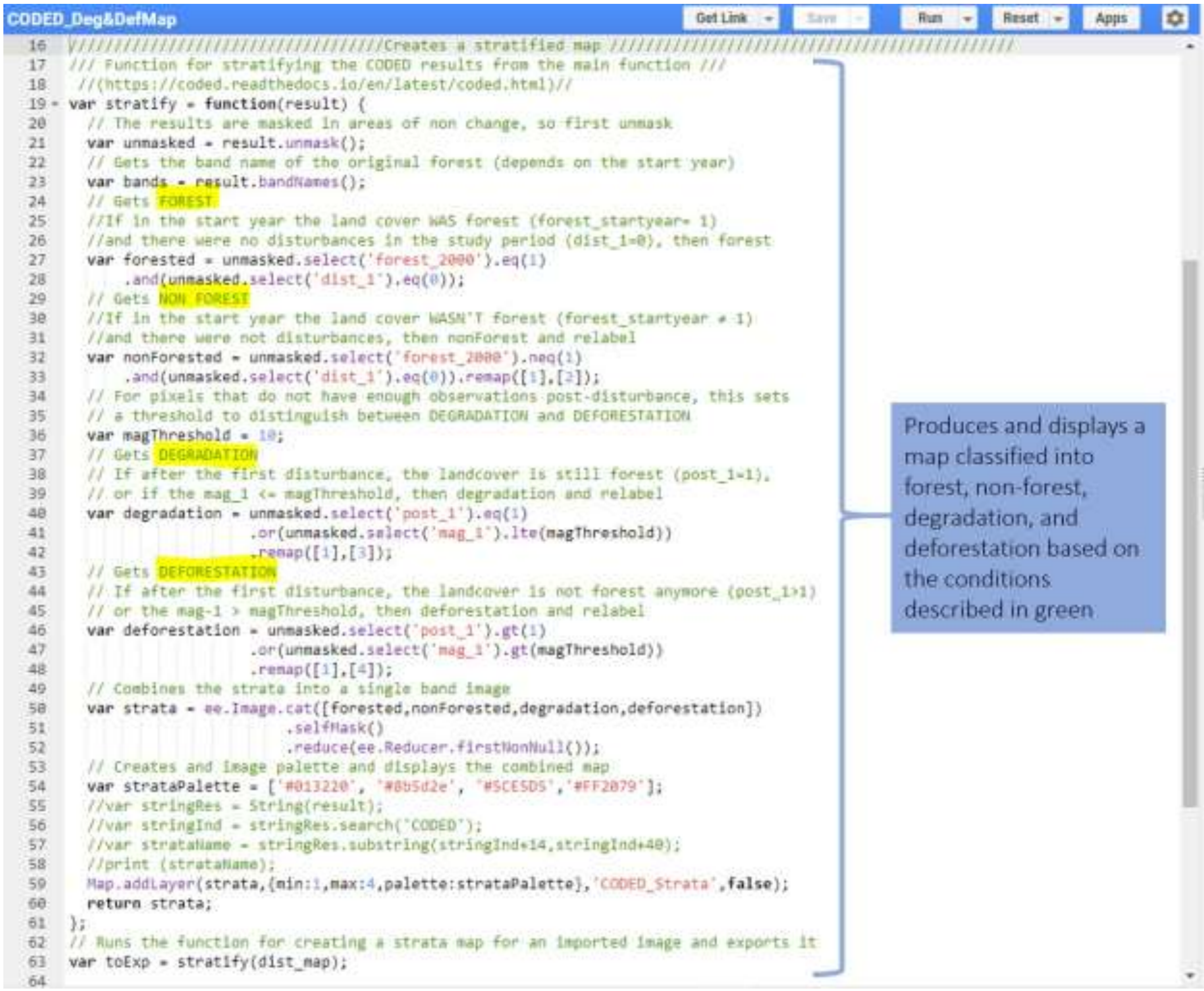
Insumos

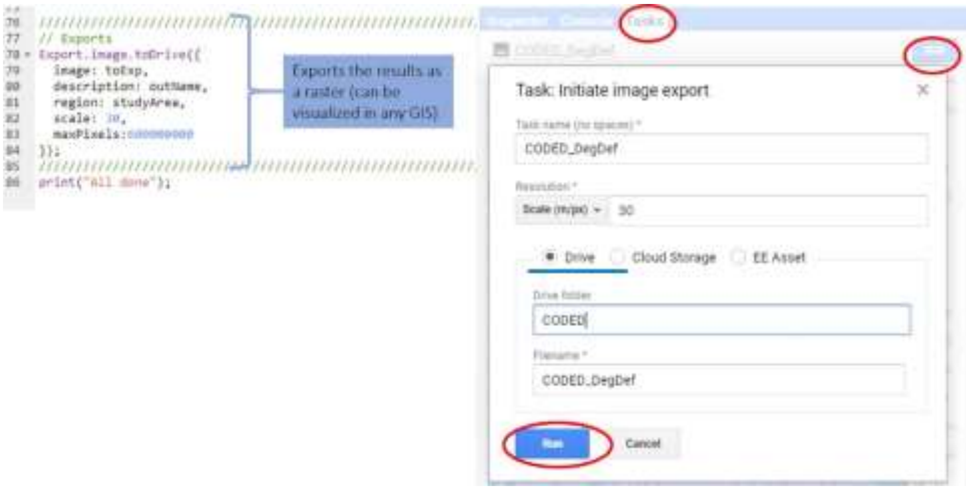
Há três entradas que o usuário tem que entrar:



Executando o Código

Execute o código clicando em run. Uma vez terminado, vá até o console de tarefas e clique em run para exportar os resultados como um raster.

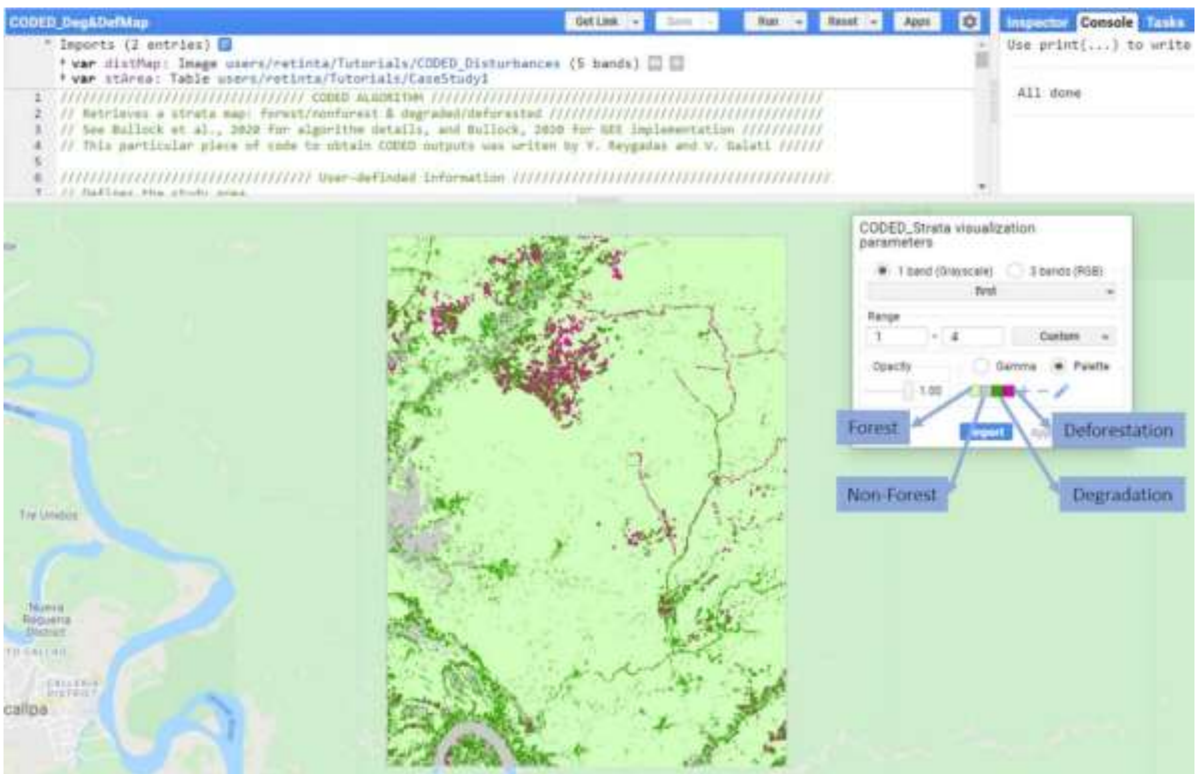




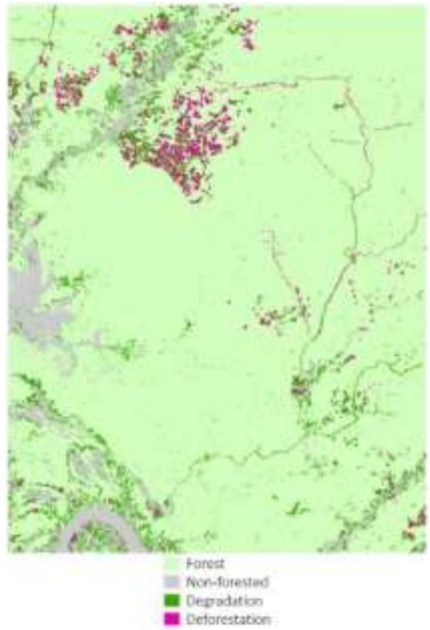
Saída

Este código produz um mapa classificado em floresta, nãofloresta, degradação e desmatamento.

Mapa exibido em GEE



Raster (".tif" file)



3.2. LandTrendr

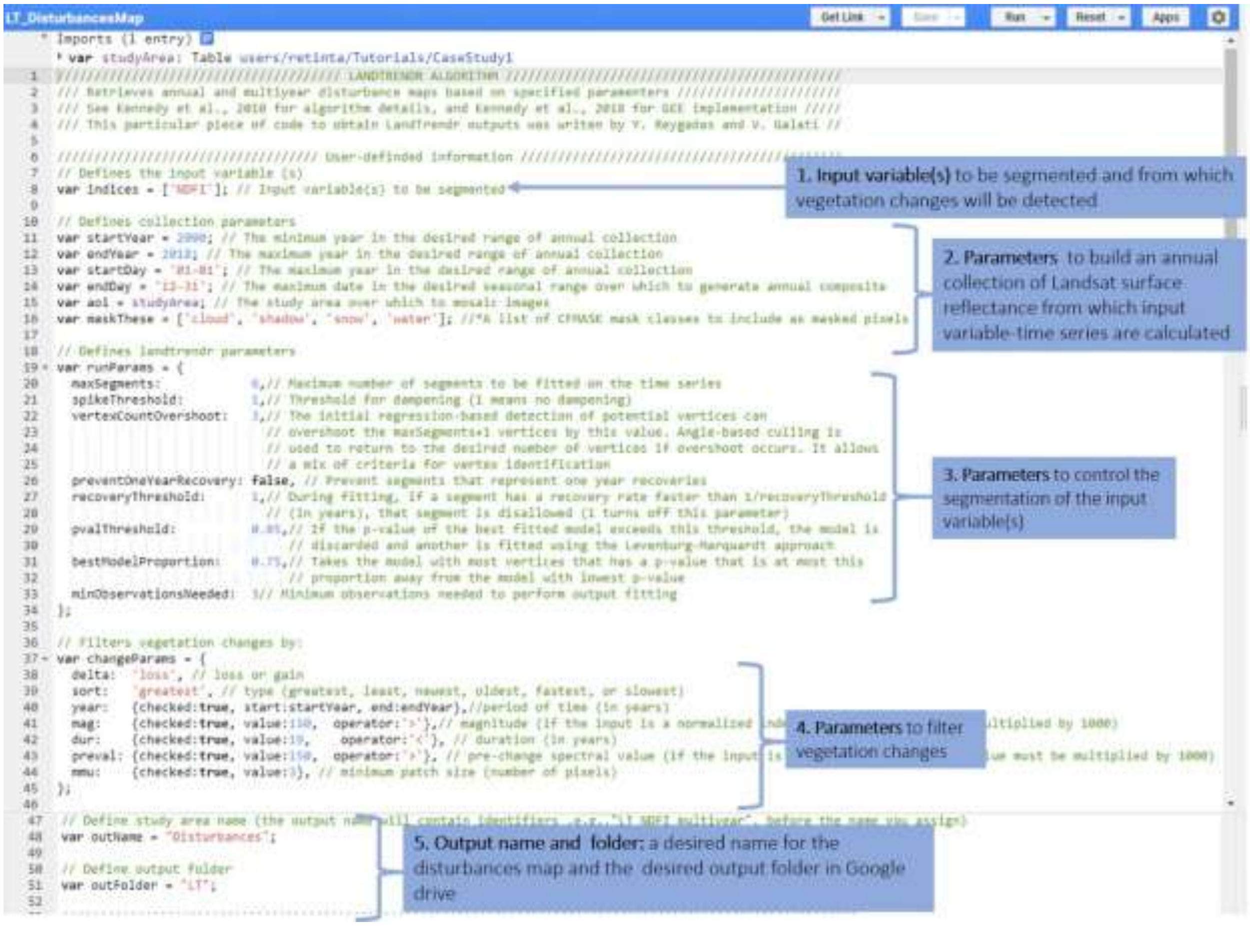
LandTrendr detecta e caracteriza a perda ou ganho de vegetação segmentando e ajustando as trajetórias temporais de uma variável definida pelo usuário (possíveis variáveis de entrada: NDFI*added, NBR, NDVI, NDSI, NDMI, TCB, TCG, TCW, TCA, e Landsat TM - bandas equivalentes 1-5 e 7). Veja Kennedy et al. (2010) para detalhes do algoritmo e Kennedy et al. (2018) para implementação do GEE.

3.2.1. Mapa de distúrbios

Esta seção demonstra como executar este código para produzir um mapa de distúrbios florestais.

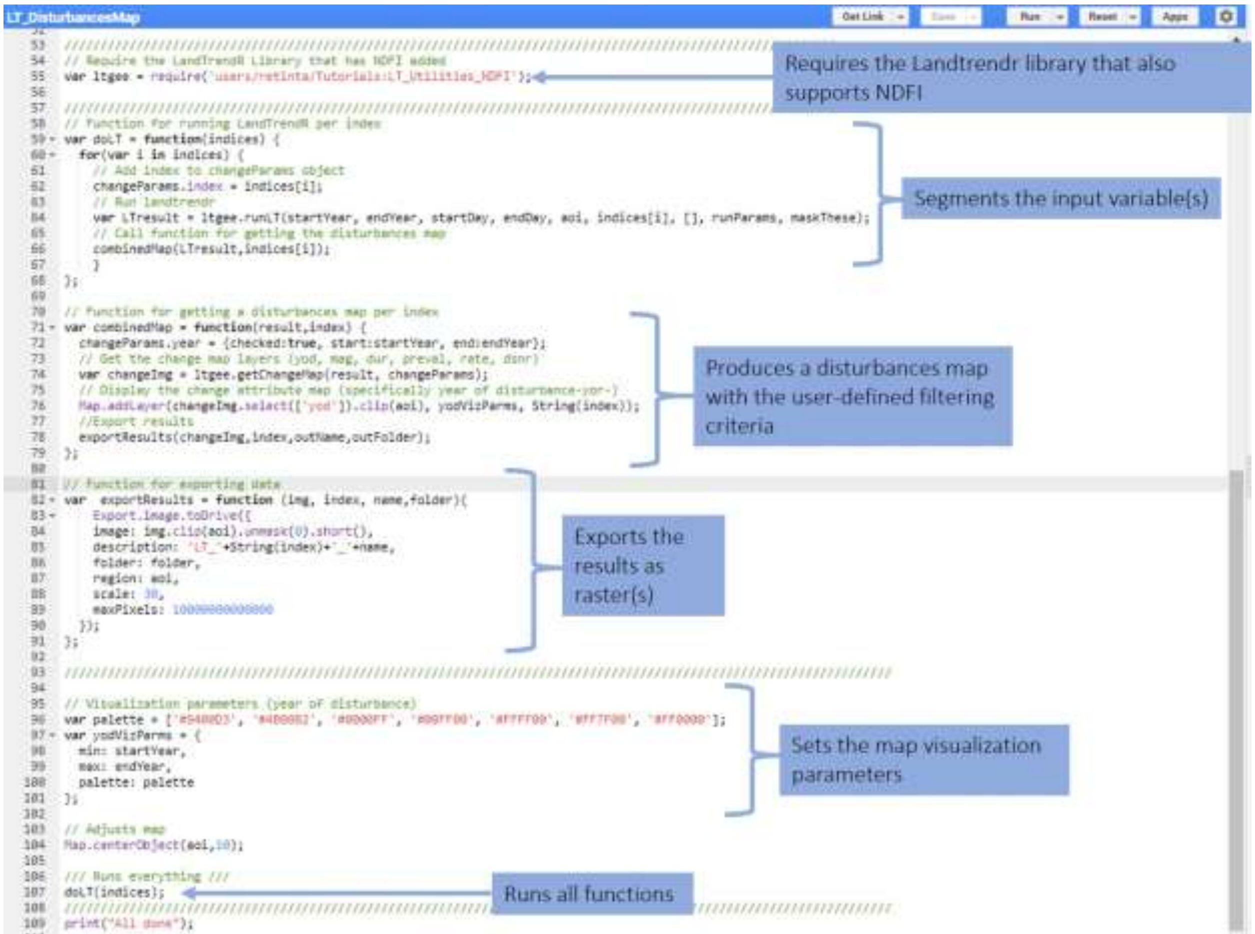
Insumos

O usuário tem que inserir as seguintes entradas:



Executando o Código

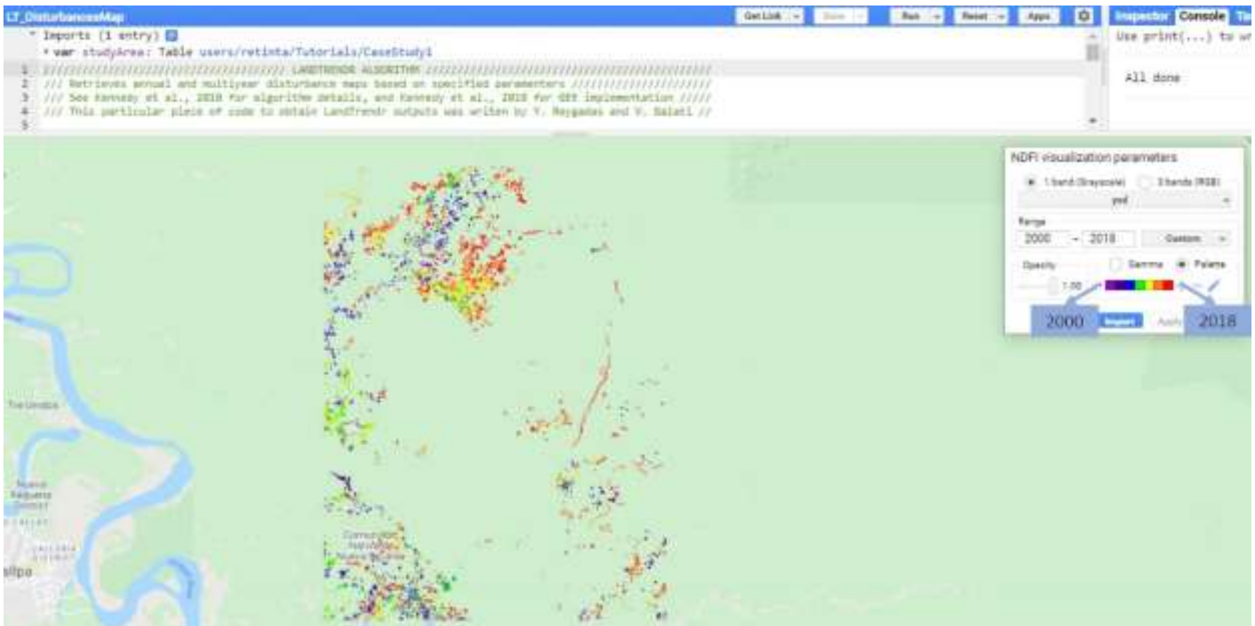
Execute o código clicando em run. Uma vez terminado, vá até o console de tarefas e clique em run para exportar os resultados como um raster.



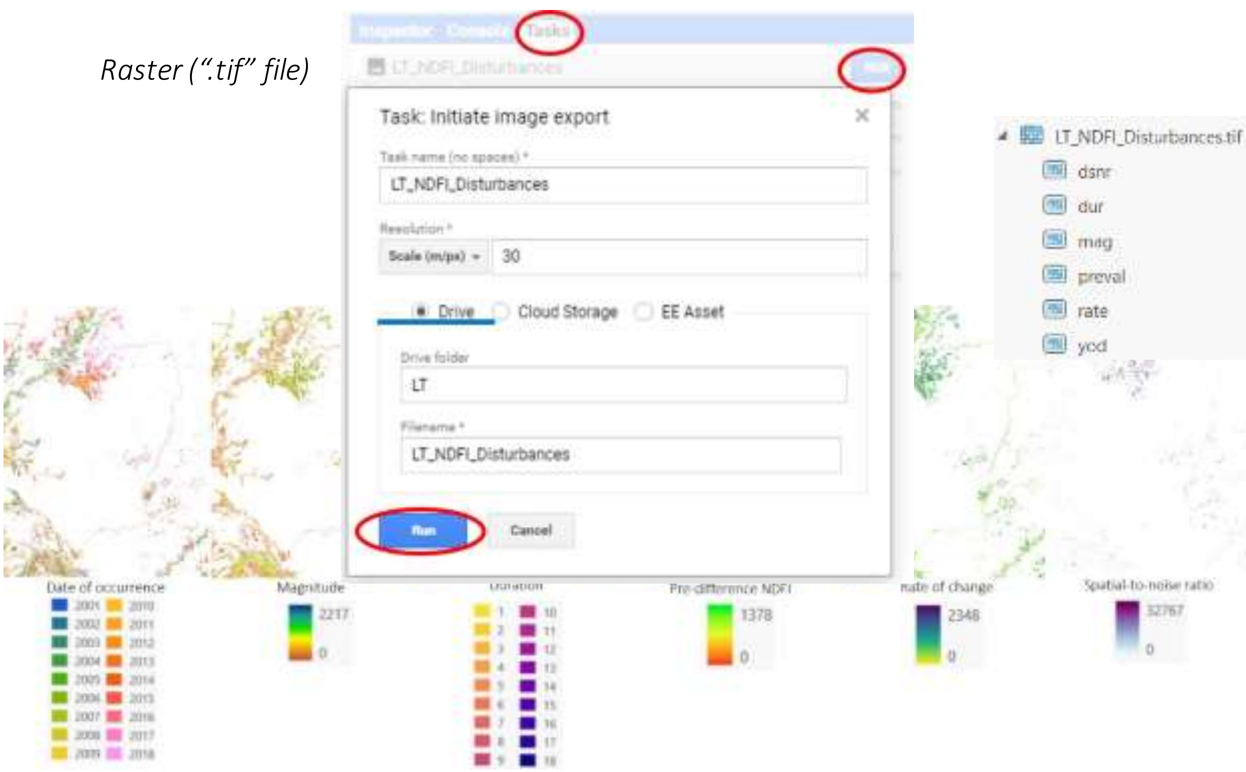
Saída

Este código produz um mapa de perturbações com seis camadas: data de ocorrência, magnitude da mudança, duração, valor espectral antes da mudança, taxa de mudança e relação sinal/ruído.

Mapa exibido em GEE (apenas a camada "ano de perturbação")



Raster (".tif" file)



3.3. MTDD

Este algoritmo baseado no MTDD classifica inicialmente as áreas florestais em intactas, degradadas e desmatadas através do treinamento de um modelo florestal aleatório com sessenta e seis métricas derivadas de seis séries temporais anuais (ou seja, NDVI, duas regiões espectrais SWIR, dois índices NDWI e SAVI) a partir das quais são calculadas onze estatísticas descritivas (ou seja, mínimo, máximo, intervalo, média, desvio padrão, coeficiente de variação, curtose, inclinação, inclinação, inclinação máxima de 5 anos e valor mais recente). Construímos este código MTDD GEE com base em Wang et al. (2019) e o adaptamos ao SWA.

3.3.1. Amostras de treinamento

Esta seção demonstra como executar este código para produzir amostras que são posteriormente utilizadas para treinar um classificador florestal aleatório.

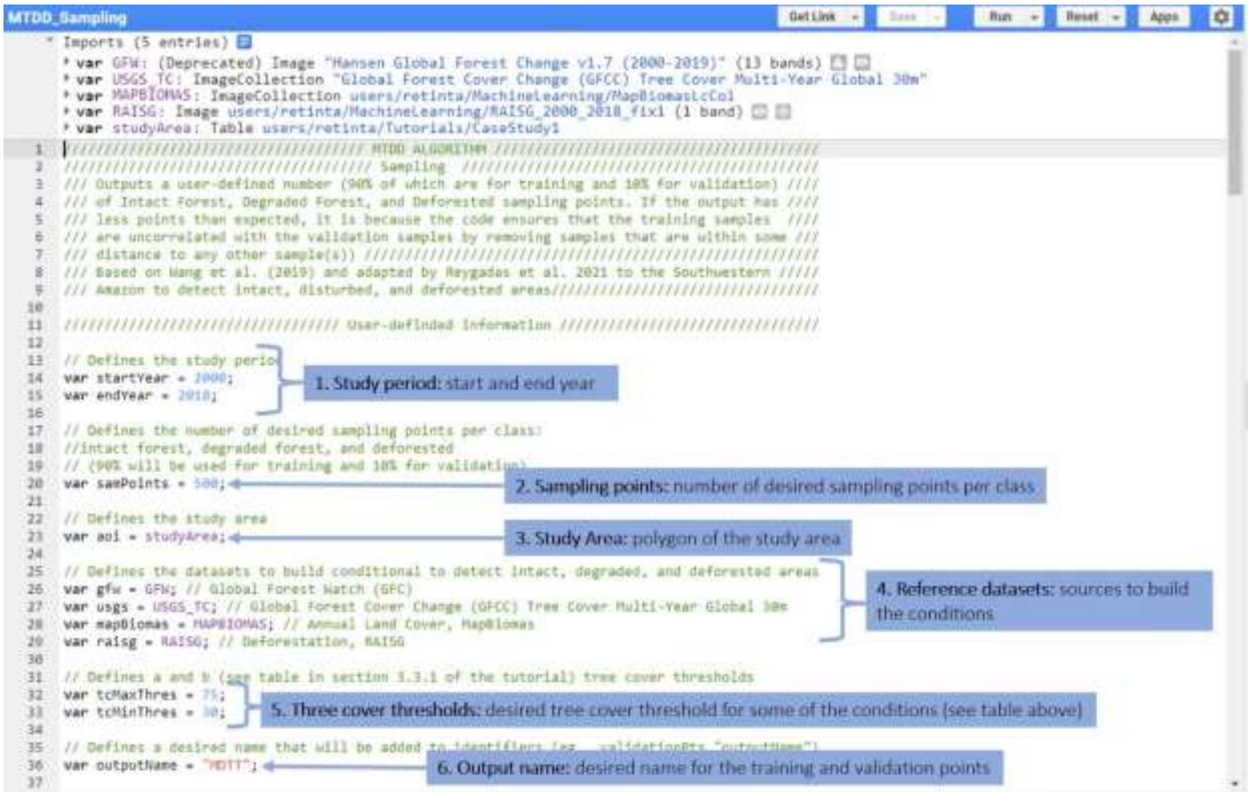
As amostras são selecionadas aleatoriamente a partir do seguinte conjunto de condições:

Classe	Condição	Dados de referência
Intacta	Cobertura florestal em todo o período de estudo	MapBiomas (MapBiomas, 2020)
	Nenhum desmatamento em todo o período de estudo	RAISG (RAISG, 2020)
	Nenhuma perda florestal em todo o período de estudo	GFC (Hansen et al., 2013)
	Cobertura de árvores superior a 75%a em 2000	GFC (Hansen et al., 2013)
	Cobertura de árvores superior a 75%a em 2015	GFCC (Sexton et al., 2013)
Degradada	Cobertura florestal no final do ano	MapBiomas (MapBiomas, 2020)
	Desmatadas em algum momento do período de estudo	RAISG (RAISG, 2020)
	Cobertura de árvores superior a 30%b e inferior a 75%a em 2015	GFCC (Sexton et al., 2013)
Deforestada	Nenhuma cobertura florestal no final do ano	MapBiomas (MapBiomas, 2020)
	Desmatadas em algum momento do período de estudo	RAISG (RAISG, 2020)
	Perda florestal em algum momento do período de estudo	GFC (Hansen et al., 2013)

[a] e [b] limiares de cobertura de árvores podem ser modificados no código GEE

Insumos

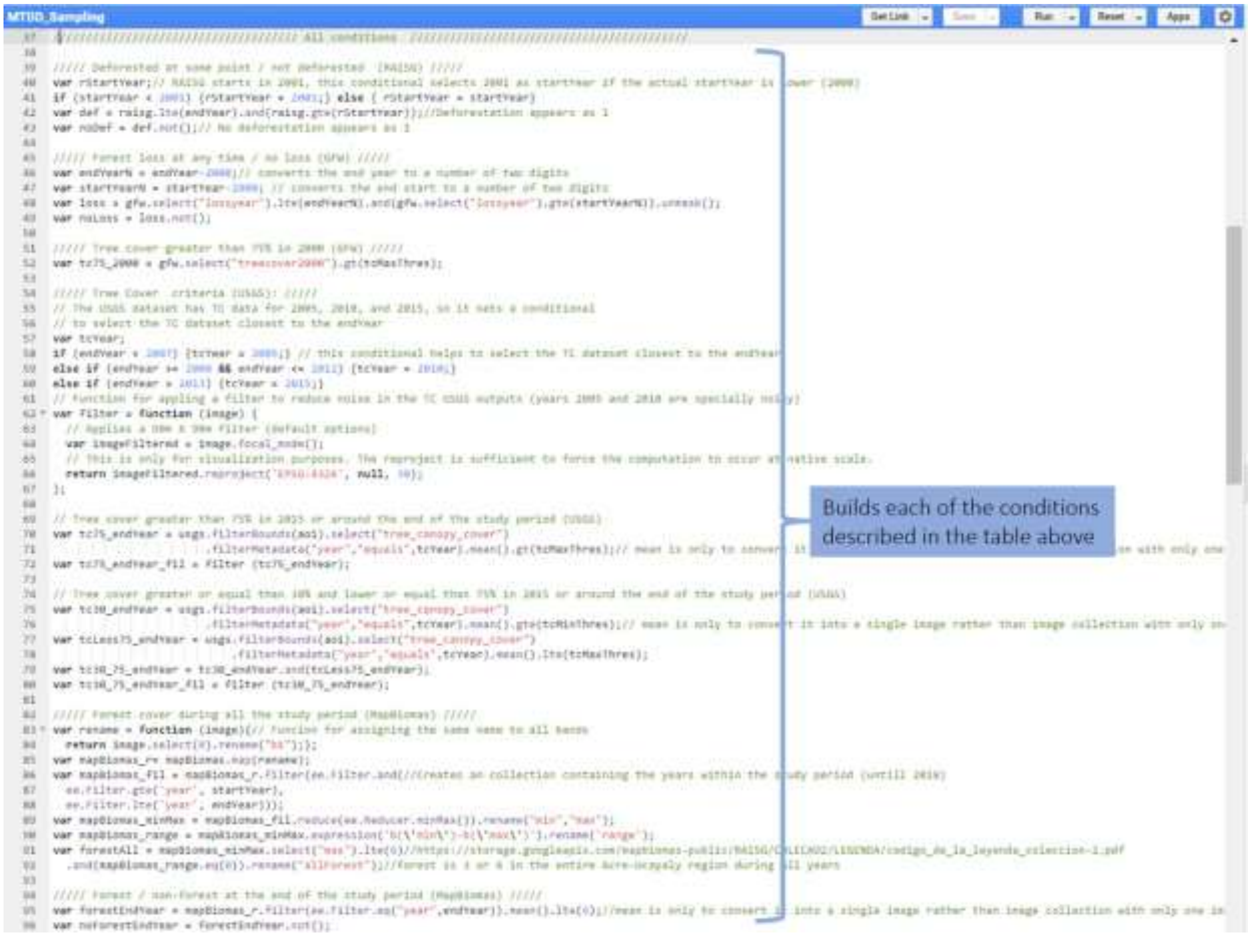
O usuário tem que inserir as seguintes entradas:

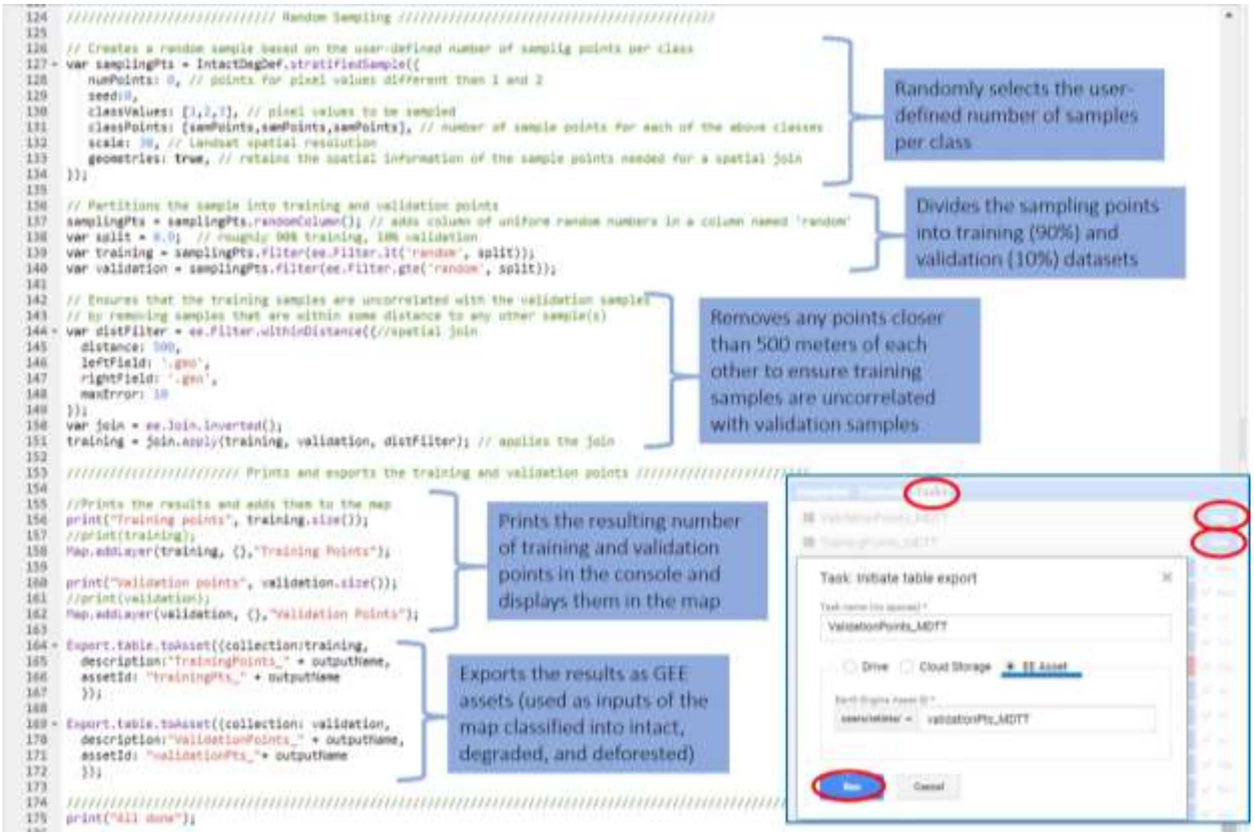


Nota: Alguns dos conjuntos de dados usados para construir as condições datam de 2018. Portanto, o período de estudo tem que cair entre 2000 e 2018.

Executando o Código

Execute o código clicando em run. Uma vez terminado, vá até o console de tarefas e clique em run para exportar os resultados como um raster.

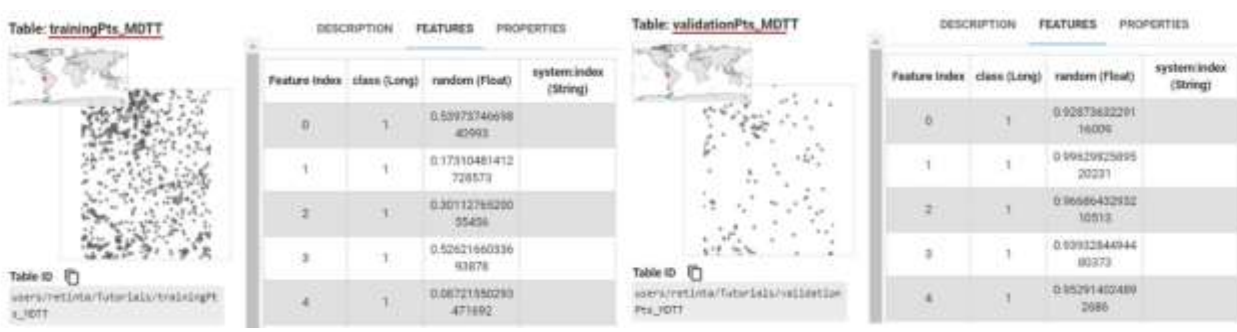




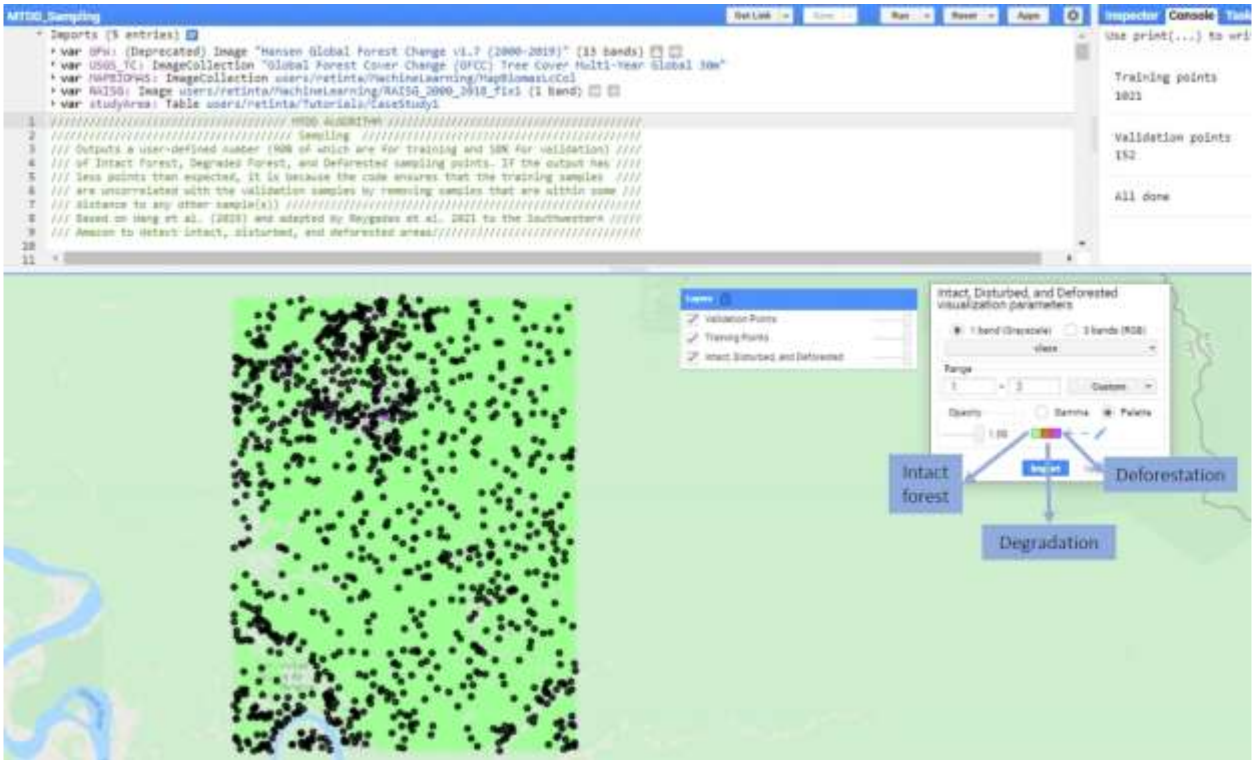
Saída

Este código produz dois conjuntos de dados: pontos de treinamento e pontos de validação.

GEE ativos



Mapa exibido em GEE

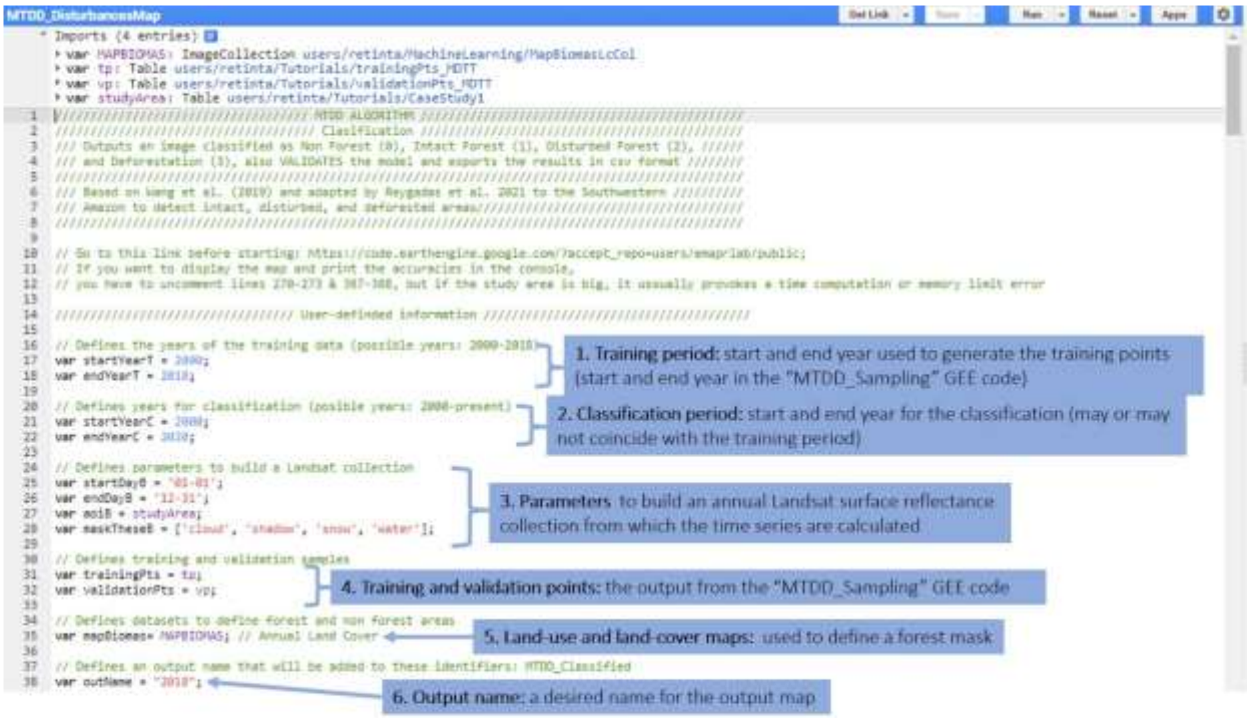


3.3.2. Mapa de degradação e desmatamento

Esta seção demonstra como executar este código para produzir um mapa classificado em floresta intacta, degradação e desmatamento.

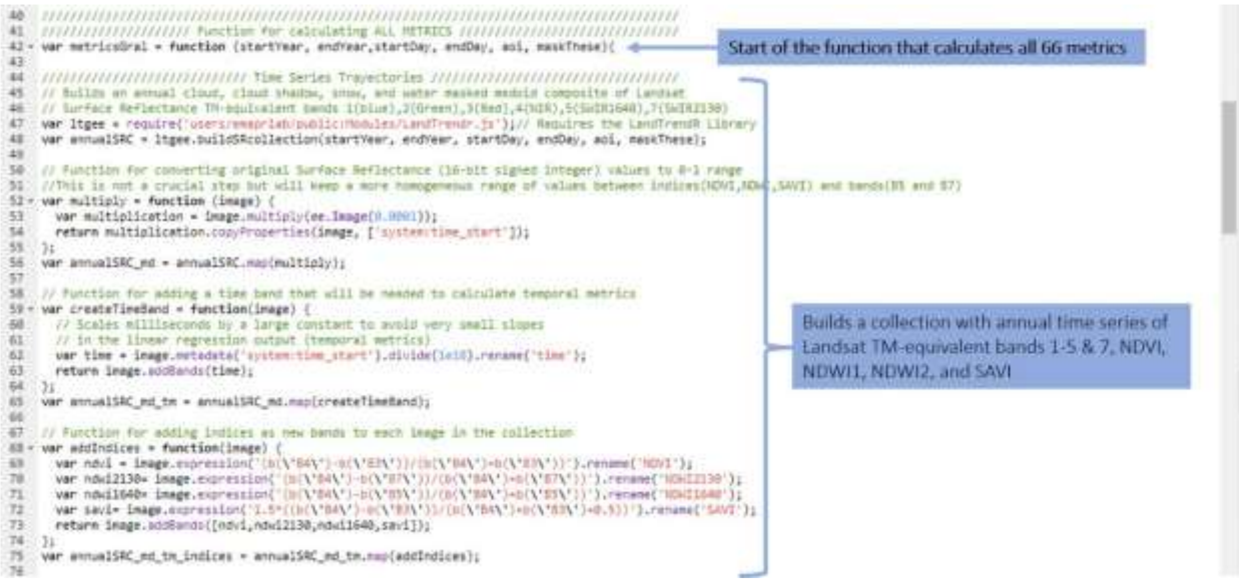
Insumos

O usuário tem que inserir as seguintes entradas:



Executando o Código

Execute o código clicando em run. Uma vez terminado, vá até o console de tarefas e clique em run para exportar os resultados como um arquivo raster (mapa classificado) e csv (matriz geral de precisão e erro).



```
/////////////////////////////////////////////////////////////////////////////////////////
/////////////////////// Function for calculating ALL METRICS ///////////////////////////////
var metricsGral = function (startYear, endYear,startDay, endDay, aoi, maskThese){

////////////////////////////// Time Series Trajectories ////////////////////////////////////
// Builds an annual cloud, cloud shadow, snow, and water masked medoid composite of Landsat
// Surface Reflectance TM-equivalent bands 1(blue),2(Green),3(Red),4(NIR),5(SWIR1640),7(SWIR2130)
var ltgee = require('users/emaprlab/public:Modules/LandTrendr.js');// Requires the LandTrendR Library
var annualSRC = ltgee.buildSRcollection(startYear, endYear, startDay, endDay, aoi, maskThese);

// Function for converting original Surface Reflectance (16-bit signed integer) values to 0-1 range
//This is not a crucial step but will keep a more homogeneous range of values between indices(NDVI,NDWI,SAVI) and bands(B5 and B7)
var multiply = function (image) {
  var multiplication = image.multiply(ee.Image(0.0001));
  return multiplication.copyProperties(image, ['system:time_start']);
};
var annualSRC_md = annualSRC.map(multiply);

// Function for adding a time band that will be needed to calculate temporal metrics
var createTimeBand = function(image) {
  // Scales milliseconds by a large constant to avoid very small slopes
  // in the linear regression output (temporal metrics)
  var time = image.metadata('system:time_start').divide(1e18).rename('time');
  return image.addBands(time);
};
var annualSRC_md_tm = annualSRC_md.map(createTimeBand);

// Function for adding indices as new bands to each image in the collection
var addIndices = function(image) {
  var ndvi = image.expression('(b(\'B4\')-b(\'B3\'))/(b(\'B4\')+b(\'B3\'))').rename('NDVI');
  var ndwi2130= image.expression('(b(\'B4\')-b(\'B7\'))/(b(\'B4\')+b(\'B7\'))').rename('NDWI2130');
  var ndwi1640= image.expression('(b(\'B4\')-b(\'B5\'))/(b(\'B4\')+b(\'B5\'))').rename('NDWI1640');
  var savi= image.expression('1.5*((b(\'B4\')-b(\'B3\'))/(b(\'B4\')+b(\'B3\')+0.5))').rename('SAVI');
  return image.addBands([ndvi,ndwi2130,ndwi1640,savi]);
};
var annualSRC_md_tm_indices = annualSRC_md_tm.map(addIndices);
```

```
///////////////////////////////// Descriptive statistics /////////////////////////////////////////

///// Location metrics: min, max, range, mean /////
var min = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.min());
var max = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.max());
var range = max.subtract(min).rename(['B5_range','B7_range','NDVI_range','NDWI2130_range','NDWI1640_range','SAVI_range']);
var mean = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.mean());

///// Scale metrics: stdDev, C.V., kurtosis, skewness /////
var stdDev = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.stdDev());
var cv = mean.divide(stdDev).rename(['B5_cv','B7_cv','NDVI_cv','NDWI2130_cv','NDWI1640_cv','SAVI_cv']);
var kurt = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.kurtosis());
var skew = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
    .reduce(ee.Reducer.skew());

///// Temporal metrics /////
/// Slope (outpus two bands: 'offset' (y-intercept) and 'scale' (slope))
var slopeB5 = annualSRC_md_tm_indices.select(['time','B5'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('B5_slp');
var slopeB7 = annualSRC_md_tm_indices.select(['time','B7'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('B7_slp');
var slopeNDVI = annualSRC_md_tm_indices.select(['time','NDVI'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('NDVI_slp');
var slopeNDWI2130 = annualSRC_md_tm_indices.select(['time','NDWI2130'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('NDWI2130_slp');
var slopeNDWI1640 = annualSRC_md_tm_indices.select(['time','NDWI1640'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('NDWI1640_slp');
var slopeSAVI = annualSRC_md_tm_indices.select(['time','SAVI'])
    .reduce(ee.Reducer.linearFit()).select('scale').rename('SAVI_slp');

/// Max-slope (maximum absolute linear regression slope of 5-year windows)
// Creates a moving 5-year window
var join = ee.Join.saveAll({//Returns a join that pairs each element from the annual SR collection
    matchesKey: 'images'    //with a group of matching elements from the 5-year window collection.
});                         //The list of matches is added to each result as an additional property
var diffFilter = ee.Filter.maxDifference({ // Establishes the 5-year filter
  difference: 78892380000, // 2.5 years in milliseconds, the filter selects 2 years before and after the target year
  leftField: 'system:time_start', // therefore, the extreme windows are made of only 3-4 years
  rightField: 'system:time_start'
});
var fiveWindowJoin = join.apply({// Collection in which each image is associated with the 5-year window images (see properties:
  primary: annualSRC_md_tm_indices,
  secondary: annualSRC_md_tm_indices,
  condition: diffFilter
});
// Function for calculating the absolute maximum slope over 5-year moving windows
var maxAbsMovSlope = function (invar, outvar){
  // Calculates slopes over 5-year windows
  var mvSlp = ee.ImageCollection(fiveWindowJoin.map(function(image) {
    var annualSRC_md_tm_indices = ee.ImageCollection.fromImages(image.get('images'));
    return ee.Image(image).addBands(annualSRC_md_tm_indices.select(["time",invar])
    .reduce(ee.Reducer.linearFit()));
  }));
  // Gets rid of the extreme windows, which are made of only 3 or 4 years
  var range1 = mvSlp.reduceColumns(ee.Reducer.minMax(), ["system:time_start"]); //Gets the date range of images in the collect
  var filter1 = mvSlp.filter(ee.Filter.date(range1.get('min')).not());
  var filter2 = filter1.filter(ee.Filter.date(range1.get('max')).not());
  var range2 = filter2.reduceColumns(ee.Reducer.minMax(), ["system:time_start"]);
  var filter3 = filter2.filter(ee.Filter.date(range2.get('min')).not());
  var filter4 = filter3.filter(ee.Filter.date(range2.get('max')).not());// Print this to see all 5-yer window slopes
  // Converts slope values to absolute numbers
  var abs = function (image) {return image.select('scale').abs()};
  // Extracts the maximum absolute slope
  return filter4.map(abs).reduce(ee.Reducer.max()).rename(outvar);
};
// Applies the absolute moving slope function to all variables
var mxSlpB5 = maxAbsMovSlope('B5', 'B5_mxSlp');
var mxSlpB7 = maxAbsMovSlope('B7','B7_mxSlp');
var mxSlpNDVI = maxAbsMovSlope('NDVI','NDVI_mxSlp');
var mxSlpNDWI2130 = maxAbsMovSlope('NDWI2130','NDWI2130_mxSlp');
var mxSlpNDWI1640 = maxAbsMovSlope('NDWI1640','NDWI1640_mxSlp');
var mxSlpSAVI = maxAbsMovSlope('SAVI','SAVI_mxSlp');

//Value at last year
var rangeDates = annualSRC_md_tm_indices.reduceColumns(ee.Reducer.minMax(), ["system:time_start"]);
var lastYearFil = annualSRC_md_tm_indices.filter(ee.Filter.date(rangeDates.get('max')));
var lastYear = lastYearFil.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI']).toBands();
var lastYearVal= lastYear.rename(['B5_last','B7_last','NDVI_last','NDWI2130_last','NDWI1640_last','SAVI_last']);

///// All metrics /////
// Creates a collection from all metrics
var metricsCollection = ee.ImageCollection([min,max,range,mean,stdDev,cv,kurt,skew,
  slopeB5,slopeB7,slopeNDVI,slopeNDWI2130,slopeNDWI1640,slopeSAVI,
  mxSlpB5,mxSlpB7,mxSlpNDVI,mxSlpNDWI2130,mxSlpNDWI1640,mxSlpSAVI,lastYearVal]);

// Converts the metrics collection to a single multi-band image
var metrics= metricsCollection.toBands();
return metrics;
};
///////////////////////////////////////////////////////////////////////////////////////////
```

Calculates 11 descriptive statistics (i.e., minimum, maximum, range, mean, standard deviation, coefficient of variation, kurtosis, skewness slope, maximum 5-year slope, and most recent value) for each of the 6 time series (i.e., NDVI, two SWIR spectral regions, two NDWI indices, and SAVI)

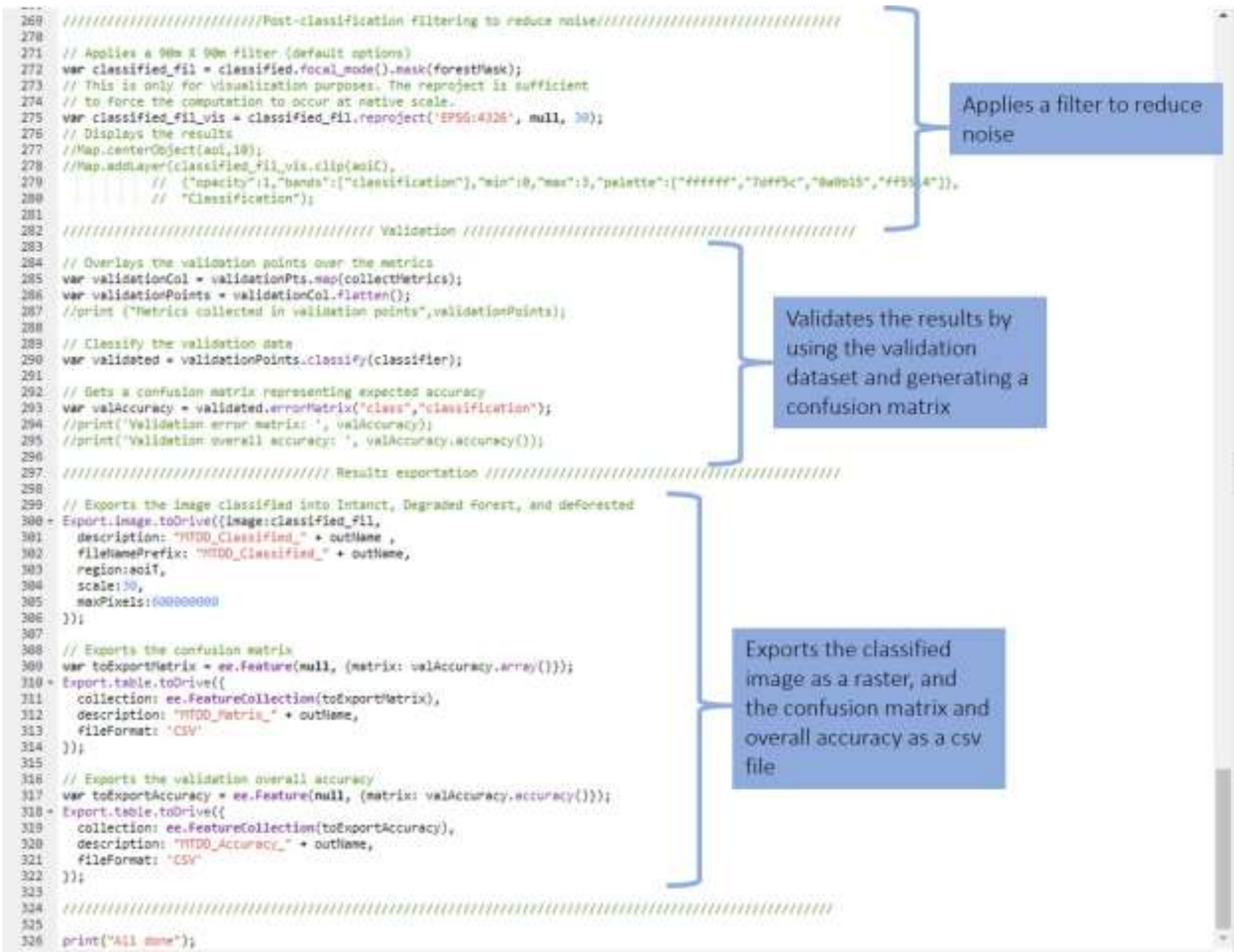
End of the function that calculates all 66 metrics

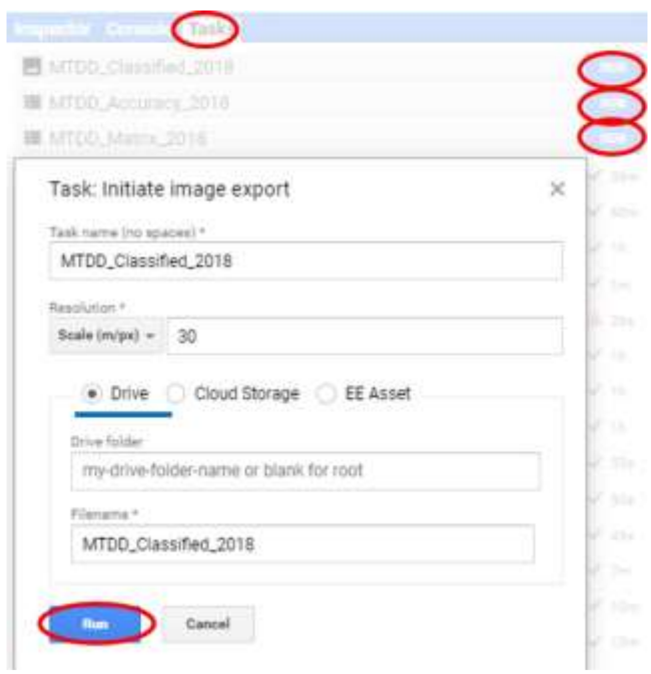
```javascript
/////////////////// Masking: delimitation of forested areas ///////////////////////////////

//// Selects areas that were forest at leat 3 consecutive years during the study period///
var rename = function (image){// Funcion for assigning the same name to all bands
  return image.select(0).rename("b1");};
var mapBiomas_r= mapBiomas.map(rename);
var mapBiomas_fil = mapBiomas_r.filter(ee.Filter.and(//Creates an collection containing the years within the study period (untill 2018)
  ee.Filter.gte('year', startYearT),
  ee.Filter.lte('year', endYearT)));
// Function for detecting pixels covered by forest
//https://storage.googleapis.com/mapbiomas-public/RAISG/COLECAO2/LEGENDA/codigo_de_la_leyenda_coleccion-2.pdf
//forest is 3 or 6 in the entire Acre-Ucayaly region during all years
var forest = function (image) {
  return image.eq(3).or(image.eq(6)).rename("forest").copyProperties(image, ['year']));
var forestCol = mapBiomas.map(forest);

// Function for adding a time band that will be needed to calculate land covers per year
var createTimeBandMask = function(image) {
  var time = image.metadata('year').rename('time');
  return image.addBands(time);
};
var forestCol_tm = forestCol.map(createTimeBandMask);

/// Forest in three consecutive years
// Creates a moving 3-year window
var join1 = ee.Join.saveAll({matchesKey: 'images'});
var diffFilter1 = ee.Filter.maxDifference({difference: 1.5, leftField: 'year', rightField: 'year'});
var threeWindowJoin = join1.apply({primary: forestCol_tm, secondary: forestCol_tm, condition: diffFilter1});

// Function for detecting forest cover over 3-year moving windows
var forest3years = function (invar, outvar){
  // Calculates sum over 3-year windows
  var f3y = ee.ImageCollection(threeWindowJoin.map(function(image) {
    var forestCol_tm = ee.ImageCollection.fromImages(image.get('images'));
    return ee.Image(image).addBands(forestCol_tm .select(["time",invar])
    .reduce(ee.Reducer.sum()));
  }));
  // Extracts the maximum sum
  return f3y.reduce(ee.Reducer.max());
};

// Applies the "detecting forest cover over 3-year" function to forest presence per year
var MaxForest3years = forest3years('forest', 'forest_3years').select("forest_sum_max");
// Gets the forest mask (areas that were forest at least three consecutive years)
var forestMask= MaxForest3years.gte(3);
```

Creates a forest mask composed of all areas that were forest al least three consecutive years during the study period

```javascript
////////////////////////////////// Classificador random forest ////////////////////////////////////////////

// *Crea una collección con todas las métricas con base en los años del muestreo
var metricsTraining = metricsGral(startYearT, endYearT,startDayB, endDayB, aoiB, maskTheseB);
print("Métricas muestreo", metricsTraining);

// **Crea una colleción con todas las métricas con base en los años para la clasificación
var metricsClassification = metricsGral(startYearC, endYearC,startDayB, endDayB, aoiB, maskTheseB);
print("Métricas clasificación", metricsClassification);

// Sobrepone los puntos de entrenamiento con las métricas*
var collectMetrics = function (feature){// usar una funcion permite a GEE trabajar con más puntos antes de exceder su capacidad
    return metricsTraining.sampleRegions({collection:feature,
    properties: ["class"],
    scale: 30,
    tileScale: 2});
};
var trainingCol = trainingPts.map(collectMetrics);
var trainingPoints = trainingCol.flatten();

// Crea un classificador random forest con 500 árboles y lo entrena
var classifier = ee.Classifier.smileRandomForest({numberOfTrees: 50})
  .train({features: trainingPoints, classProperty:"class",
  inputProperties:['0_B5_min','0_B7_min','0_NDVI_min','0_NDWI2130_min','0_NDWI1640_min','0_SAVI_min',
    '1_B5_max','1_B7_max','1_NDVI_max','1_NDWI2130_max','1_NDWI1640_max','1_SAVI_max',
    '2_B5_range','2_B7_range','2_NDVI_range','2_NDWI2130_range','2_NDWI1640_range','2_SAVI_range',
    '3_B5_mean','3_B7_mean','3_NDVI_mean','3_NDWI2130_mean','3_NDWI1640_mean','3_SAVI_mean',
    '4_B5_stdDev','4_B7_stdDev','4_NDVI_stdDev','4_NDWI2130_stdDev','4_NDWI1640_stdDev','4_SAVI_stdDev',
    '5_B5_cv','5_B7_cv','5_NDVI_cv','5_NDWI2130_cv','5_NDWI1640_cv','5_SAVI_cv',
    '6_B5_kurtosis','6_B7_kurtosis','6_NDVI_kurtosis','6_NDWI2130_kurtosis','6_NDWI1640_kurtosis','6_SAVI_kurtosis',
    '7_B5_skew','7_B7_skew','7_NDVI_skew','7_NDWI2130_skew','7_NDWI1640_skew','7_SAVI_skew',
    '8_B5_slp','9_B7_slp','10_NDVI_slp','11_NDWI2130_slp','12_NDWI1640_slp','13_SAVI_slp',
    '14_B5_mxSlp','15_B7_mxSlp','16_NDVI_mxSlp','17_NDWI2130_mxSlp','18_NDWI1640_mxSlp','19_SAVI_mxSlp',
    '20_B5_last','20_B7_last','20_NDVI_last','20_NDWI2130_last','20_NDWI1640_last','20_SAVI_last']

  });

// Classifica las métricas deseadas**
var metricsMask = metricsClassification.mask(forestMask);// Enmascara las areas no forestales
var classified = metricsMask.classify(classifier);//No bosque (0), bosque intacto (1), bosque degradado (2), áreas deforestadas (3)
print("Mapa clasificado");
```

Entrena un clasificador *random forest* con las 66 métricas de los datos de entrenamiento y la clasifica las imágenes deseadas

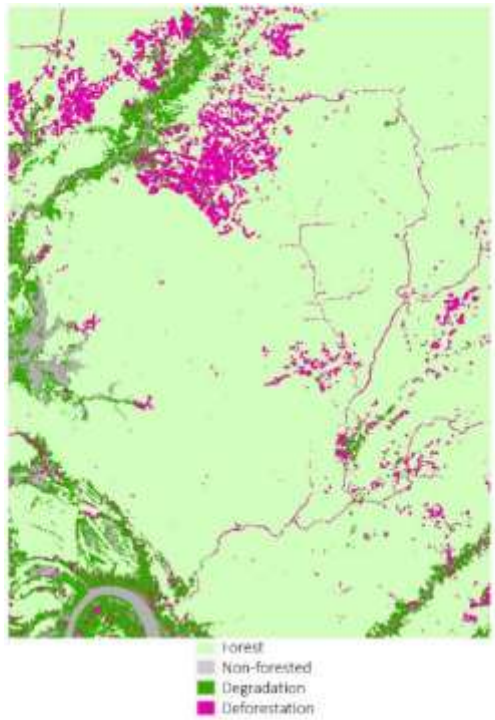

//////////////////////////////Post-classification filtering to reduce noise//////////////////////////////////
// Applies a 90m X 90m filter (default options)
var classified_fil = classified.focal_mode().mask(forestMask);
// This is only for visualization purposes. The reproject is sufficient
// to force the computation to occur at native scale.
var classified_fil_vis = classified_fil.reproject('EPSG:4326', null, 30);
// Displays the results
//Map.centerObject(aoi,10);
//Map.addLayer(classified_fil_vis.clip(aoiC),
// {"opacity":1,"bands":["classification"],"min":0,"max":3,"palette":["ffffff","7dff5c","0a0b15","ff5_4"]},
// "Classification");
// Validation //
// Overlays the validation points over the metrics
var validationCol = validationPts.map(collectMetrics);
var validationPoints = validationCol.flatten();
//print ("Metrics collected in validation points",validationPoints);
// Classify the validation data
var validated = validationPoints.classify(classifier);
// Gets a confusion matrix representing expected accuracy
var valAccuracy = validated.errorMatrix("class","classification");
//print('Validation error matrix: ', valAccuracy);
//print('Validation overall accuracy: ', valAccuracy.accuracy());
/////////////////////////////////////// Results exportation //
// Exports the image classified into Intanct, Degraded Forest, and deforested
Export.image.toDrive({image:classified_fil,
description: "MTDD_Classified_" + outName ,
fileNamePrefix: "MTDD_Classified_" + outName,
region:aoiT,
scale:30,
maxPixels:600000000
});
// Exports the confusion matrix
var toExportMatrix = ee.Feature(null, {matrix: valAccuracy.array()});
Export.table.toDrive({
collection: ee.FeatureCollection(toExportMatrix),
description: "MTDD_Matrix_" + outName,
fileFormat: 'CSV'
});
// Exports the validation overall accuracy
var toExportAccuracy = ee.Feature(null, {matrix: valAccuracy.accuracy()});
Export.table.toDrive({
collection: ee.FeatureCollection(toExportAccuracy),
description: "MTDD_Accuracy_" + outName,
fileFormat: 'CSV'
});
print("All done");
Applies a filter to reduce noise
Validates the results by using the validation dataset and generating a confusion matrix
Exports the classified image as a raster, and the confusion matrix and overall accuracy as a csv file


Task
MTDD_Classified_2018
MTDD_Accuracy_2018
MTDD_Matrix_2018
Task: Initiate image export
Task name (no spaces) *
MTDD_Classified_2018
Resolution *
Scale (m/px)
30
Drive
Cloud Storage
EE Asset
Drive folder
my-drive-folder-name or blank for root
Filename *
MTDD_Classified_2018
Run
Cancel

Saída

Este código produz um mapa classificado em floresta não florestal, floresta intacta, degradação e desmatamento.

Raster (".tif" file)



Matriz geral de precisão e confusão (".csv" arquivos)

	A	B	C
1	system:in	matrix	.geo
2	0	0.927632	
3			

	A	B	C	D	E	F
1	system:in	matrix	.geo			
2	0	[[0, 0, 0, 0], [0, 60, 0, 0], [0, 0, 47, 4], [0, 1, 6, 34]]				
3						
4						

Referências

Bullock, E., 2020. Continuous Degradation Detection (CODED) — coded 0.2 documentation [WWW Document]. URL https://coded.readthedocs.io/en/latest/old.html (accessed 2.9.21).

Bullock, E.L., Woodcock, C.E., Olofsson, P., 2020. Monitoring tropical forest degradation using spectral unmixing and Landsat time series analysis. Remote Sens. Environ. 238, 110968. https://doi.org/10.1016/j.rse.2018.11.011

Cohen, W.B., Healey, S.P., Yang, Z., Stehman, S. V., Brewer, C.K., Brooks, E.B., Gorelick, N., Huang, C., Hughes, M.J., Kennedy, R.E., Loveland, T.R., Moisen, G.G., Schroeder, T.A., Vogelmann, J.E., Woodcock, C.E., Yang, L., Zhu, Z., 2017. How Similar Are Forest Disturbance Maps Derived from Different Landsat Time Series Algorithms? Forests 8, 98. https://doi.org/10.3390/f8040098

FAO, 2001. Global Forest Resources Assessment 2000. Rome, Italy.

Hansen, M.C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S.A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S.J., Loveland, T.R., Kommareddy, A., Egorov, A., Chini, L., Justice, C.O., Townshend, J.R.G., 2013. High-resolution global maps of 21st-century forest cover change. Science (80-.). 342, 850–853. https://doi.org/10.1126/science.1244693

Kennedy, R.E., Yang, Z., Cohen, W.B., 2010. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr-Temporal segmentation algorithms. https://doi.org/10.1016/j.rse.2010.07.008

Kennedy, R.E., Yang, Z., Gorelick, N., Braaten, J., Cavalcante, L., Cohen, W.B., Healey, S., 2018. Implementation of the LandTrendr algorithm on Google Earth Engine. Remote Sens. 10, 691. https://doi.org/10.3390/rs10050691

MapBiomas, 2020. MapBiomas Amazonia Project-Collection 2.0 of annual land-cover and land-use maps [WWW Document]. URL https://code.earthengine.google.com/toolkit-download

McDowell, N.G., Coops, N.C., Beck, P.S.A., Chambers, J.Q., Gangodagamage, C., Hicke, J.A., Huang, C. ying, Kennedy, R., Krofcheck, D.J., Litvak, M., Meddens, A.J.H., Muss, J., Negrón-Juarez, R., Peng, C., Schwantes, A.M., Swenson, J.J., Vernon, L.J., Williams, A.P., Xu, C., Zhao, M., Running, S.W., Allen, C.D., 2015. Global satellite monitoring of climate-induced vegetation disturbances. Trends Plant Sci. https://doi.org/10.1016/j.tplants.2014.10.008

RAISG, 2020. Amazon Geo-Referenced Socio-Environmental Information Network [WWW Document]. URL https://www.amazoniasocioambiental.org/ (accessed 3.30.21).

Reygadas, Y., Spera, S., Galati, V., Salisbury, D.S., Silva, S., Novoa, S., 2021. Mapping Forest

Disturbances Across the Southwestern Amazon: Evaluation and Comparison of Optical Remote Sensing Algorithms. Remote Sens. Environ. Under review.

Sasaki, N., Putz, F.E., 2009. Critical need for new definitions of “forest” and “forest degradation” in global climate change agreements. Conserv. Lett. 2, 226–232. https://doi.org/10.1111/j.1755-263X.2009.00067.x

Schoene, D., Killmann, W., von Lüpke, H., LoycheWilkie, M., 2007. Definitional Issues Related to Reducing Emissions from Deforestation in Developing Countries, Vol. 5. ed. Food and Agriculture Organization of the United Nations, Rome.

Sexton, J.O., Song, X.-P., Feng, M., Noojipady, P., Anand, A., Huang, C., Kim, D.-H., Collins, K.M., Channan, S., Dimiceli, C., Townshend, J.R., 2013. International Journal of Digital Earth Global, 30-m resolution continuous fields of tree cover: Landsat-based rescaling of MODIS vegetation continuous fields with lidar-based estimates of error. Int. J. Digit. Earth 6, 427–448. https://doi.org/10.1080/17538947.2013.786146

Wang, Y., Ziv, G., Adami, M., Mitchard, E., Batterman, S.A., Buermann, W., Schwantes Marimon, B., Marimon Junior, B.H., Matias Reis, S., Rodrigues, D., Galbraith, D., 2019. Mapping tropical disturbed forests using multi-decadal 30 m optical satellite imagery. Remote Sens. Environ. 221, 474–488. https://doi.org/10.1016/j.rse.2018.11.028