Diversidade de solo

## Importando pacotes e inicializando *geemap*

import os  
import ee  
import geemap

geemap.ee\_initialize()

A diversidade de solo é incluida na diversidade da paisagem em locais com baixa variedade de *landforms*, amplitude de elevação e *wetland score*. A diversidade de solos é calculada somando o número de tipos de solo presentes na vizinhaça da célula focal.

## Banda de Dados

Nós utilizamos a base de dados de pedologia do Instituto Brasileiro de Geografia e Estatística (IBGE) como base para o cálculo da diversidade de solos. Os polígonos foram posteriomente filtrados por tipo de solo e rasterizados na escala do modelo digital de elevação que usamos nas etapas anteriores (~ 90 metros).

## Códigos para a criação da diversidade de solos

As análises foram rodadas no *Google Earth Engine* (Gorelick 2017), devido a demanda computacional do projeto, usando o pacote **geemap** (Wu 2020), em *Python* (Python Software Foundation 2023), como interface pela facilidade na documentação e reprodutividade das análises.

O primeiro passo é criar uma função que filtra os polígonos pelo nome do solo, atribui valor 1 ao polígono, transforma em *raster* e reprojeta para a resolução do modelo digital de elevação que estamos usando em todas as análises (~90 metros). Depois, a função cria uma imagem cujas bandas são as células da vizinhaça da célula focal. Células com a presença do solo tem valor 1 e sem o solo 0. Ao somarmos os valores das células dentro da vizinhança temos o número de células com o tipo de solo. Por fim, transformamos valores maior que 0 em 1 para classificarmos o *raster* como presença ou ausência do tipo de solo na vizinhança da célula focal.

# Função para mudar o valor da propriedade do polígono para 1  
def set\_feature(feature):  
 return feature.set("cod\_simbol", 1)  
  
# Função para filtrar o polígono por tipo de solo,   
# transformar em raster e retornar um raster de presença  
# e ausência do solo na vizinhança   
def soil\_presence(code):  
 # Filtra o tipo de solo e atribui valor 1 (para rasterizar)   
 type = (soil.filter(ee.Filter.eq("cod\_simbol", code))  
 .map(set\_feature))  
   
 # Transforma o polígono em raster  
 soil\_rast = type.reduceToImage(\*\*{  
 'properties': ["cod\_simbol"],  
 'reducer': ee.Reducer.first()  
 })  
  
 # Converte o raster para a escala do modelo digital de elevação  
 soil\_raster = (soil\_rast  
 .reproject(\*\*{'crs': "EPSG:4326",  
 'scale': escala}))  
  
 radius\_pixels = 15  
  
 # Conte o número de células na vizinhança que possuem o tipo de solo selecionado  
 soil\_count = (soil\_raster.neighborhoodToBands(ee.Kernel.circle(radius\_pixels))  
 .reduce(ee.Reducer.count()))  
  
 # Tranforme os valore maiores que 0 em 1, indicando a presença  
 #ou ausência do tipo de solo na vizinhança  
 soil\_diversity = soil\_count.where(soil\_count.gt(0), 1)  
  
 return soil\_diversity

# Importando mapa de biomas do IBGE para extrair as coordenadas mínimas e máximas do Brasil  
bioma = ee.FeatureCollection("projects/ee-lucasljardim9/assets/Biome")  
  
def func\_cmp(feature):  
 return feature.bounds()   
  
# Extraindo as coordenadas mínimas e máximas do Brasil  
bioma\_box = bioma.map(func\_cmp).geometry().dissolve(\*\*{'maxError': 1}).bounds()  
  
# Extraindo a resolução do mapa  
DEM = ee.Image("MERIT/DEM/v1\_0\_3")   
  
escala = DEM.projection().nominalScale()

Nós importamos os polígonos de solo do IBGE e filtramos a coluna com o tipo de solo (“cod\_simbol”), e criamos uma lista com os nomes dos tipos de solo, eliminando os polígonos que não tinham nomes.

# Importando os polígonos de solo do IBGE e selecionado a coluna com tipos de solo  
soil = (ee.FeatureCollection("projects/ee-lucasljardim9/assets/soil\_IBGE")  
 .select("cod\_simbol"))  
  
# Criando uma lista com os nomes dos tipos de solo  
soil\_list = soil.reduceColumns(ee.Reducer.toList(), ["cod\_simbol"]).values().get(0)  
  
# Criando uma lista com os nomes de solo, sem duplicatas  
soil\_codes = (ee.List(soil\_list).distinct()  
 .filter(ee.Filter.neq("item", "")))

Depois, aplicamos a função que criamos anteriormente para cada nome de solo e transformamos em bandas de uma imagem. Ao somarmos os valores de cada célula nas bandas calculamos o ńumero de tipos de solo presentes na vizinhança de cada célula focal, a diversidade de solos.

# Crie o raster de presença e ausencia para cada tipo de solo  
soil\_list = soil\_codes.map(soil\_presence)  
  
# Transforme os raster de presença de solo em bandas de   
# uma imagem e soma para calcular a diversidade de solos  
# Converta do doble para permitir a exportação do dado  
soil\_diversity = (ee.ImageCollection.fromImages(soil\_list)  
 .toBands()  
 .reduce("sum")  
 .toDouble())

Por fim, exportamos o *raster* como um *asset* no *Google Earth Engine*.

# Exporte o raster de diversidade de solo   
assetId = "projects/ee-lucasljardim9/assets/soil\_diversity"  
  
geemap.ee\_export\_image\_to\_asset(  
 soil\_diversity, description='soil\_diversity', assetId=assetId, region=bioma\_box, scale=escala,maxPixels=1e13  
)