

Diversidade de solo

Importando pacotes e inicializando *geemap*

```
import os
import ee
import geemap

geemap.ee_initialize()
```

A diversidade de solo é incluída 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 vizinhança 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 posteriormente 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 vizinhança 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()))

    # Transforma os valores 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()

# Extraíndo as coordenadas mínimas e máximas do Brasil
bioma_box = bioma.map(func_cmp).geometry().dissolve(**{'maxError': 1}).bounds()

# Extraíndo 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 número de tipos de solo presentes na vizinhança de cada célula focal, a diversidade de solos.

```

# Crie o raster de presença e ausência 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 double para permitir a exportação do dado
soil_diversity = (ee.ImageCollection.fromImages(soil_list)
                  .toBands()
                  .reduce("sum"))

```

```
        .toDouble())

# 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=
)
```