Amplitude de Elevação

Importando pacotes e inicializando geemap

```
import os
import ee
import geemap
```

```
geemap.ee_initialize()
```

A amplitude de elevação foi calculada como a diferença entre a elevação máxima e mínima dentro da vizinhaça da célula focal. Como essa métrica é correlacionada com a variedade de *landforms*, nós calculamos o resíduos de uma regressão (Ordinary Least Squares) entre as duas váriaveis. Assim, a amplitude de elevação residual é independente da variedade de *landforms*, permitindo a identificação de locais que tenham maior variabilidade microclimática que a proporcionada pela variedade de *landforms*, quando compormos o índice de diversidade da paisagem.

Base de Dados

Nós utilizamos o Modelo Digital de Elevação (DEM) do Merit-DEM (Yamazaki et al. 2017), na escala de 90 metros. O Merit-DEM é um produto global que combina dados dos satélites do Shuttle Radar Topography Mission (SRTM) (Farr et al. 2007) e Advanced Land Observing Satellite (ALOS) (Tadono et al. 2014), permitindo a replicabilidade da metodologia em outras regiões. Além disso, o Merit-DEM corrige viéses de Modelo Digitais de Elevação gerados por imagens de satétite como *speckle noise*, *stripe noise*, *absolute bias* e *tree height bias* (Yamazaki et al. 2017). A correção de *tree height bias* é principalmente importante para a Floresta Amazônica devido à sua densidade de árvores altas.

A variedade de *landforms* foi calculada anteriormente (veja o capítulo Variedade de *Landforms*) e está disponível com *asset* em **"projects/ee-lucasljardim9/assets/landform_variety"**.

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.

Códigos para o cálculo da amplitude de elevação residual

Nossas 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.

```
# Importando mapa de biomas do IBGE para extrair as coordenadas mínimas e máximas do Brasi
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()
```

Nós importamos os *rasters* do modelo digital de elevação e da variedade de *landforms*.

```
# Importando o modelo digital de elevação
DEM = ee.Image("MERIT/DEM/v1_0_3")

# Importando a variedade de landforms calculada anteriormente
landform_variety = ee.Image("projects/ee-lucasljardim9/assets/landform_variety")

#Escala dos rasters ~92 metros
escala = DEM.projection().nominalScale()
```

Nós extraímos do *raster* as células dentro da vizinhança (kernel circular com 5 células de raio, ~450 metros) da célula focal e salvamos como bandas de uma imagem. Assim, cada banda é um *stack* das células da vizinhança da célula focal, a primeira banda possui todas as primeiras células de cada célula focal, a segunda banda todas as segunda células e assim por diante.

```
# Tamanho do raio do kernel para o calculo da amplitude de elevação
radius_pixels = 5

# Criando rasters da vizinhança de cada célula como bandas da imagem
neighbor = DEM.neighborhoodToBands(ee.Kernel.circle(ee.Number(radius_pixels)))
```

A imagem *neighbor* criada anteriormente as células da vizinhança como bandas da imagem. Assim, as primeiras células de cada banda são as células da vizinhança da primeira célula focal, organizadas como colunas (bandas). Ao calcularmos os valores máximos e mínimos para cada coluna de *neighbor*, estamos calculando os valores máximos e mínimos da vizinhança de cada célula focal.

```
# Calcule o máximo da vizinhança
elevation_max = neighbor.reduce(ee.Reducer.max())

# Calcule o mínimo da vizinhança
elevation_min = neighbor.reduce(ee.Reducer.min())
```

Subtraindo os valores máximos e mínimos de cada célula focal e calculando o seu valor absoluto, temos a amplitude de elevação para cada célula focal. Nós salvamos a amplitude de elevação e variedade de *landforms* como uma imagem com duas bandas, sendo a primeira banda a variável preditora da regressão e a segunda banda a variável resposta.

```
# Calcule a amplitude da vizinhança
elevation_range = elevation_max.subtract(elevation_min).abs()

# Crie uma imagem com as bandas de varidedade de landforms e amplitude de elevação
# A primeira imagem é o x da regressão e a segunda é o y

elevation = (ee.Image.cat(landform_variety, elevation_range)
            .rename(['landform_variety', 'elevation_range']))
```

Desta forma, aplicamos a regressão entre as variáveis.

```
# Rode uma regressão linear (OLS) entre varidade de landforms e amplitude de elevação
regression = elevation.reduceRegion(**{
     'reducer': ee.Reducer.linearFit(),
     'geometry': bioma_box,
     'maxPixels': 1e13,
     'scale': escala
    })
```

Após a regressão, multiplicamos a variável preditora pelo coeficiente de regressão (*slope*) e adicionamos o valor do intercepto para predizermos os valores de amplitude de elevação esperados pela regressão. Em seguida, subtraimos os valores de amplitude de elevação pelos valores preditos pela regressão para calcularmos os resíduos do modelo.

```
# Calcule o valor predito, pela regressão, de amplitude elevação, sem intercepto
pred = elevation.select('landform_variety').multiply(ee.Number(regression.get('scale')))

# Adicione o intercepto na predição
predict = pred.add(ee.Number(regression.get('offset')))

# Calcule o residuo da regressão
residuals = elevation.select('elevation_range').subtract(predict).rename(['residuals'])
```

Por fim, exportamos o raster de amplitude de elevação residual como um *asset* do Google Earth Engine.

```
# Exporte a amplitude de elevação residual como asset
assetId = "projects/ee-lucasljardim9/assets/elevetion_range_residual"

geemap.ee_export_image_to_asset(
    residuals,
    description='elevation_range_residual',
    assetId=assetId,
    region=bioma_box,
    scale=escala,
    maxPixels=1e13
)
```