Diversidade da paisagem

## Importando pacotes e inicializando *geemap*

import os  
import ee  
import geemap

geemap.ee\_initialize()

A diversidade da paisagem é uma métrica composta pelos valores de Z da variedade de *landforms*, amplitude de elevação, *wetland score* e diversidade de solos. Inicialmente, a diversidade da paisage assume os valores de variedade de *landforms*. Nos locais onde a amplitude de elevação é maior que variedade de *landforms*, a diversidade da paisagem é substituída pela média ponderada entre as duas variáveis, atribuindo peso dobrado para a variedade de *landforms*. Locais onde wetland score é maior que a diversidade da paisagem calculada anteriormente, os valores são substituídos por wetland score. Na localidades onde a diversidade de solo é maior que a diversidade da paisagem, os valores são substituídos pela média ponderada das variáveis naquela localidade. A última etapa é truncar os valores de Z que são *outliers* para deixar a distribuição de diversidade da paisagem mais suavizada. A suavização é importante para que locais com alta diversidade da paisagem, devido a sua especificidade de variedade de landforms e amplitude da elevação não possuam mais peso no mapa de resiliência que os locais guiados por *wetland score*.

## Base de Dados

Os dados para a diversidade da paisagem são os criados na etapa de cálculo dos *Z-scores*, descrito anteriormente.

## Códigos para a criação da diversidade da paisagem

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.

O primeiro passo é criar uma função para truncar a diversidade da paisagem por uma porcentagem (**coverage**) de cobertura da distribuição dos valores.

def truncate\_z\_scores(image, coverage):  
   
 # Calcule os percentis baseado na cobertura definida em coverage  
 percentile = [100 - coverage, coverage]  
   
 # Encontre os valores dos percentis definidos  
 quartiles = image.reduceRegion(\*\*{'reducer': ee.Reducer.percentile(percentile),  
 'scale': image.projection().nominalScale(),  
 'maxPixels':1e13});  
 #5% quantiles  
 q5 = quartiles.getNumber('sum\_p5')  
 #95% quantiles  
 q95 = quartiles.getNumber('sum\_p95')  
  
 # Truncar a imagem pelo percentis   
 truncated\_z = (image.where(image.lt(q5), q5)  
 .where(image.gt(q95), q95))  
  
 return truncated\_z

Nós importamos os dados de bioma para delimitar a região de análise e o raster do Brasil para que os mapas sejam cortados para o Brasil antes do cálculo da diversidadade da paisagem.

# Importando o polígono de bioma para definir as   
# coordenadas máximas e mínimas do Brasil  
  
bioma = ee.FeatureCollection("projects/ee-lucasljardim9/assets/Biome")  
  
brasil\_raster = ee.Image("projects/ee-lucasljardim9/assets/brasil\_raster")  
  
# ModeloDigital de Elevação para extrair a resolução  
DEM = ee.Image("MERIT/DEM/v1\_0\_3")  
  
# função para extrair as bordas dos polígonos  
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()  
  
# Resolução das análises  
escala = DEM.projection().nominalScale()

Importamos os mapas e cortamos para o raster do Brasil.

# Importando e cortando as imagens para o Brasil  
landform\_Z = (ee.Image('projects/ee-lucasljardim9/assets/Z\_landform\_variety\_byregion')  
 .updateMask(brasil\_raster))  
  
elevation\_range\_Z = (ee.Image('projects/ee-lucasljardim9/assets/Z\_elevation\_range\_byregion')  
 .updateMask(brasil\_raster))  
  
wetland\_score = (ee.Image('projects/ee-lucasljardim9/assets/Z\_wetlands\_score\_byregion')  
 .updateMask(brasil\_raster))  
  
soil\_diversity = (ee.Image('projects/ee-lucasljardim9/assets/Z\_soil\_diversity\_byregion')  
 .updateMask(brasil\_raster))

Aqui começamos o cálculo da diversidade da paisagem, atribuindo à diversidade da paisagem os valores de variedade de *landforms*. Depois testamos os locais onde a amplitude de elevação é maior que a variedade de *landforms* e substituímos os valores.

landscape\_diversity = landform\_Z  
  
# Testando se o Z da amplitude de elevação é maior que o Z da variedade de landsforms  
test\_1 = elevation\_range\_Z.gt(landscape\_diversity)  
  
# Média ponderada entre variedade de landforms e amplitude de elevação  
average\_elevation = landform\_Z.multiply(2).add(elevation\_range\_Z).divide(3)  
  
# Substituindo os valores de variedade de landforms por amplitude de elevação   
# onde o segundo valor é maior que o primeiro  
landscape\_diversity = landscape\_diversity.where(test\_1, average\_elevation)

Agora, nós adicionamos *wetland score*, primeiro testando onde *wetland score* é maior que a diversidade da paisagem calculada anteriormente, e nos locais cujos valores de *wetland score* são maiores que a diversidade da paisagem, nós substituímos os valores.

# testando onde wetland score é maior que a diversidade da paisagem  
# com variedade de landforms e amplitude de elevação  
test\_2 = wetland\_score.gt(landscape\_diversity)  
  
# Substitua o valor de diversidade da paisagem por wetland score  
#onde o segundo é maior que o primeiro  
landscape\_diversity = landscape\_diversity.where(test\_2, wetland\_score)

A inclusão da diversidade de solos é um pouco mais complexa. Nós testamos onde a diversidade de solo é maior a diversidade da paisagem já calculada e substituímos os valores por 4 médias ponderadas diferentes, cada uma representando os passos anteriores. A primeira média é para os locais onde a diversidade de solos é maior que *wetland score*, que é maior que amplitude de elevação. Na segunda média, os valores são para as regiões com maior diversidade de solos, mas onde *wetland score* é menor que a amplitude de elevação. A terceira média é para locais onde a amplitude de elevação não é maior que variedade de *landforms*, mas possui valores menores de *wetland score*, que também são menores que a diversidade de solos. A quarta média é para locais onde somente a diversidade de solos é maior que a variedade de *landforms*.

# teste onde diversidade de solo é maior que a diversidade da paisagem  
test\_3 = soil\_diversity.gt(landscape\_diversity)  
  
# Calcule as médias ponderadas  
average\_soil\_1 = (landform\_Z.multiply(2)  
 .add(elevation\_range\_Z)  
 .add(wetland\_score)  
 .add(soil\_diversity)  
 .divide(5))  
  
average\_soil\_2 = (landform\_Z.multiply(2)  
 .add(elevation\_range\_Z)  
 .add(soil\_diversity)  
 .divide(4))  
  
average\_soil\_3 = (landform\_Z.multiply(2)  
 .add(wetland\_score)  
 .add(soil\_diversity)  
 .divide(4))  
  
average\_soil\_4 = (landform\_Z.multiply(2)  
 .add(soil\_diversity)  
 .divide(3))  
  
# Substitua os valores de diversidade da paisagem pela diversidade de solo  
# os os valores de solo são maiores  
landscape\_diversity = (landscape\_diversity  
 .where(test\_1.And(test\_2).And(test\_3), average\_soil\_1))  
  
landscape\_diversity = (landscape\_diversity  
 .where(test\_1.And(test\_2.Not()).And(test\_3), average\_soil\_2))  
  
landscape\_diversity = (landscape\_diversity  
 .where(test\_1.Not().And(test\_2).And(test\_3), average\_soil\_3))  
  
landscape\_diversity = (landscape\_diversity  
 .where(test\_1.Not().And(test\_2.Not()).And(test\_3), average\_soil\_4))

Com o mapa de diversidade da paisagem pronto, nos aplicamos um truncamento dos valores maiores e menores que 95% da distribuição dos dados, para que haja uma suavização da imagem, impedindo que *outliers* guiem os mapas posteriores de resiliência da paisagem.

truncated\_landscape\_diversity = truncate\_z\_scores(landscape\_diversity, 95)

Finalizamos exportando o mapa de diversidade da paisagem como *asset* no *Google Earth Engine*.

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