PA_GEE

September 5, 2023

0.1 Get Mha of bioenergy outside and inside protected areas within the top fractions of the landscape for biodiversity.

This script uses the Google Earth Engine Collection "World Database on Protected Areas". The collection updated regularly on GEE

[1]:
```
# import required libraries
import ee
ee.Initialize()
import geemap
Map = geemap.Map()
import pandas as pd
```

Functions

[2]:
```
def get_stats(fraction):
    # lists for loops
    continent = ['Africa','Europe','North America','Asia','South␣
 ↪America','Oceania']
    scenarios = ['ssp1','ssp2','ssp5']
    # store results in list
    res = []
    for i in scenarios:
        if fraction==17:
            frac = '17%'
            # get original mask
            mask = ee.Image('users/marcogirardello/annarepo/'+i).select('b1').
 ↪float()
            # get image with pixels outside protected areas
            opa = ee.Image('users/marcogirardello/annarepo/'+i+'_nopa').
 ↪updateMask(mask.eq(2))
            # get image with pixels inside protected areas
            inpa = ee.Image('users/marcogirardello/annarepo/'+i+'_pa').
 ↪updateMask(mask.eq(2))
        elif fraction==30:
            frac = '30%'
            # get image with pixels outside protected areas
            opa = ee.Image('users/marcogirardello/annarepo/'+i+'_nopa')
            # get image with pixels inside protected areas
```

```
            inpa = ee.Image('users/marcogirardello/annarepo/'+i+'_pa')
        # --------------- global:  outside protected areas
        valopa = opa.reduceRegion(geometry = opa.geometry(), reducer = ee.
↪Reducer.sum().unweighted(),scale = 50000, maxPixels = 1e13).getInfo()
        # store results for non-protected areas
        res.append((pd.DataFrame([valopa])
                .rename({'b1':'Mha'},axis = 1)
                .assign(coverage = 'global',scenario = i, type = 'outside␣
↪protected areas',fraction = frac)))
         # --------------- global: within protected areas
        valinpa = inpa.reduceRegion(geometry = inpa.geometry(), reducer = ee.
↪Reducer.sum().unweighted(),scale = 50000, maxPixels = 1e13).getInfo()
        # store results for protected areas
        res.append(pd.DataFrame([valinpa])
               .rename({'b1':'Mha'},axis = 1)
               .assign(coverage = 'global',scenario = i, type = 'inside␣
↪protected areas', fraction = frac))
        # --------------- by continent
        for y in continent:
            cont = countries.filterMetadata('CONTINENT','equals',y)
            # ----- outside protected areas
            opa1 = opa.clip(cont)
            opacont = opa1.reduceRegion(geometry = opa.geometry(),scale =␣
↪50000, reducer = ee.Reducer.sum().unweighted(),maxPixels = 1e13).getInfo()
            res.append(pd.DataFrame([opacont])
                       .rename({'b1':'Mha'},axis = 1)
                       .assign(coverage = y,scenario = i,type = 'outside␣
↪protected areas',fraction = frac))
            # ----- outside protected areas
            inpa1 = inpa.clip(cont)
            inpacont = inpa1.reduceRegion(geometry = inpa.geometry(),scale =␣
↪50000,reducer = ee.Reducer.sum().unweighted(), maxPixels = 1e13).getInfo()
            res.append(pd.DataFrame([inpacont])
                       .rename({'b1':'Mha'},axis = 1)
                       .assign(coverage = y, scenario = i, type = 'inside␣
↪protected areas',fraction = frac))
    return pd.concat(res,ignore_index=True)
```

Load relevant collections and images

A note on the bioenergy images. The whole image is the top 30% fraction. The top 17% fraction can be obtained by filtering with a mask (b2 = 3)

[3]:
```
# wpa dataset
wpa = ee.FeatureCollection("WCMC/WDPA/current/polygons")
# dissolve wpa data
```

```
#wpa1 = wpa.map(lambda feature: feature.buffer(30)).union()

ssp1 = ee.Image('users/marcogirardello/annarepo/ssp1')
ssp2 = ee.Image('users/marcogirardello/annarepo/ssp2')
ssp5 = ee.Image('users/marcogirardello/annarepo/ssp5')

# get continents
countries = ee.FeatureCollection('users/marcogirardello/annarepo/countries')
```

[4]:
```
wpa = wpa.filterMetadata('STATUS','equals','Designated')
```

[5]:
```
scenarios =['ssp1','ssp2','ssp5']
```

Clip and export bionenergy data

[6]:
```
for i in scenarios:
    # get image
    tmp = ee.Image('users/marcogirardello/annarepo/'+i)
    # get bioenergy data
    values = tmp.select('b2')
    # get everything inside protected areas
    values1 = values.clip(wpa)
    # get everything outside protected areas
    values2 = values1.where(values1.gt(0),1)
    values3 = values.updateMask(values2.unmask().Not())
    # outside PA
    task = ee.batch.Export.image.toAsset(image = values3,assetId='users/
 ↪marcogirardello/annarepo/'+i+'_nopa',crs='EPSG:4326',maxPixels=1e13,
                                region = values3.geometry(),scale=␣
 ↪50000,description = i)
    task.start()
    # inside PA
    task = ee.batch.Export.image.toAsset(image = values1,assetId='users/
 ↪marcogirardello/annarepo/'+i+'_pa',crs='EPSG:4326',maxPixels=1e13,
                                region = values3.geometry(),scale=␣
 ↪50000,description = i)
    task.start()
```

Get Mha of bioenergy inside and outside protected areas

[7]:
```
res = get_stats(fraction = 30)
```

[7]:
```
res1 = get_stats(fraction = 17)
```

[12]:
```
pd.concat([res,res1]).to_csv('/mnt/data1tb/Dropbox/AnnaRepo/project/
 ↪barchartstats/barchart_stats.csv',index = False)
```