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

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

Functions

In [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(ma
            # get image with pixels inside protected areas
            inpa = ee.Image('users/marcogirardello/annarepo/'+i+'_pa').updateMask(mas
        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
        # store results for non-protected areas
        res.append((pd.DataFrame([valopa])
                .rename({'b1':'Mha'},axis = 1)
                .assign(coverage = 'global',scenario = i, type = 'outside protected a
         # --------------- global: within protected areas
        valinpa = inpa.reduceRegion(geometry = inpa.geometry(), reducer = ee.Reducer.
        # store results for protected areas
        res.append(pd.DataFrame([valinpa])
               .rename({'b1':'Mha'},axis = 1)
               .assign(coverage = 'global',scenario = i, type = 'inside protected are
        # --------------- 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, redu
            res.append(pd.DataFrame([opacont])
                       .rename({'b1':'Mha'},axis = 1)
                       .assign(coverage = y,scenario = i,type = 'outside protected ar
            # ----- outside protected areas
            inpa1 = inpa.clip(cont)
            inpacont = inpa1.reduceRegion(geometry = inpa.geometry(),scale = 50000,re
            res.append(pd.DataFrame([inpacont])
```

```
                        .rename({'b1':'Mha'},axis = 1)
                        .assign(coverage = y, scenario = i, type = 'inside protected a
    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)

In [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')
```

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

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

Clip and export bionenergy data

In [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/marcogirardel
                                region = values3.geometry(),scale= 50000,description =
    task.start()
    # inside PA
    task = ee.batch.Export.image.toAsset(image = values1,assetId='users/marcogirardel
                                region = values3.geometry(),scale= 50000,description =
    task.start()
```

Get Mha of bioenergy inside and outside protected areas

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

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

In [12]:
```
pd.concat([res,res1]).to_csv('/mnt/data1tb/Dropbox/AnnaRepo/project/barchartstats/bar
```