sample_assign_depth

March 15, 2022

The Coastal Grain Size Portal (C-GRASP) dataset Will Speiser, Daniel Buscombe, Evan Goldstein > Assign Depth to Sample

The purpose of this notebook
This notebook will output a dataframe containing all of the data from a chosen C-GRASP dataset with a new field containing a depth estimation from NOAA CUDEM topobathy dataset. As both C-Grasp and CUDEM file sizes vary completion of this task will vary with internet connectivity. This notebook provides simple code that estimates a sample's depth based on CUDEM files.
To do so, a user can choose a C-GRASP and CUDEM dataset of choice. The notebook then downloads all of the CUDEM files of chosen resolution to the user's computer. Please choose resolution carefully as this process take a long time depending on which resolution the user chooses. Then the notebook converts each CUDEM cell value to a csv containing the CUDEM file's depth value and location for each cell. After, these csv's are combined into one dataframe After the CUDEM data conversion, the chosen C-GRASP dataset is downloaded and converted to a dataframe. Finally the two datasets are converted to GeoPandasData frames and are joined by proximity, assigning each downloaded CGRASP sample a depth from the nearest CUDEM .

[1]:
```
#!pip install wget
```

```
Collecting wget
  Downloading wget-3.2.zip (10 kB)
  Preparing metadata (setup.py) … done
Building wheels for collected packages: wget
  Building wheel for wget (setup.py) … done
  Created wheel for wget: filename=wget-3.2-py3-none-any.whl size=9675
sha256=fe270103d22c1dd91838ca1f4798f284525d0f6ff8980e440163fd4beb45e95c
  Stored in directory: /home/marda/.cache/pip/wheels/8b/f1/7f/5c94f0a7a505ca1c81
cd1d9208ae2064675d97582078e6c769
Successfully built wget
Installing collected packages: wget
Successfully installed wget-3.2
```

[2]:
```
import requests
from bs4 import BeautifulSoup
import netCDF4
import pandas as pd
import os
import glob
import ipywidgets
```

```
import geopandas as gpd
%matplotlib inline
import matplotlib.pyplot as plt
import wget
```

0.1 Select a dataset. Choose your CUDEM dataset mindfully as the higher resolution files can take significantly longer to download.

[26]:
```
#Dataset collection widget
zen=ipywidgets.Select(
    options=['Entire Dataset', 'Estimated Onshore Data'], #, 'Verified Onshore␣
 ↪Data', 'Verified Onshore Post 2012 Data'
    value='Entire Dataset',
    # rows=10,
    description='Dataset:',
    disabled=False
)

display(zen)

#Dataset collection widget
resc=ipywidgets.Select(
    options=['3 arc-second', '1 arc-second', '1/3 arc-second', '1/9␣
 ↪arc-second'],
    value='3 arc-second',
    # rows=10,
    description='CUDEM:',
    disabled=False
)

display(resc)
```

```
Select(description='Dataset:', options=('Entire Dataset', 'Estimated Onshore␣
 ↪Data'), value='Entire Dataset')
```

```
Select(description='CUDEM:', options=('3 arc-second', '1 arc-second', '1/3␣
 ↪arc-second', '1/9 arc-second'), val…
```

0.2 Here, we download the chosen Cudem Dataset

First we grab the appropriate abreviation to put in the download url

[27]:
```
if resc.value == '3 arc-second':
    res='3as'
if resc.value == '1 arc-second':
    res='1as'
if resc.value == '1/3 arc-second':
    res='13as'
```

```
if resc.value == '1/9 arc-second':
    res='19as'
```

0.2.1 Then we download your data to file. Depending on the resolution you pick and your download speeds this can take from minutes to nearly a day, so proceed with caution!

Here we find all the file download links

[28]:
```
def get_url_paths(url, ext='', params={}):  #function for extracting all the␣
 ↪file download link names
    response = requests.get(url, params=params)
    if response.ok:
        response_text = response.text
    else:
        return response.raise_for_status()
    soup = BeautifulSoup(response_text, 'html.parser')
    parent = [url + node.get('href') for node in soup.find_all('a') if node.
 ↪get('href').endswith(ext)]
    return parent

url = 'https://www.ngdc.noaa.gov/thredds/catalog/tiles/tiled_'+res+'/catalog.
 ↪html' #catalogue we will call them from with the appropriate resolution
ext = 'nc' #Only find the .nc CUDEM files
result = get_url_paths(url, ext) #call the scraping function
```

Here we will use all of the links found above to download all of your data.

[29]:
```
#Make links into df
link_df = pd.DataFrame(columns = ['link'])
link_df['link']=result

#split link field by '/' delimiter to get file name

link_df['filename']=link_df['link'].str.split('/', expand=True)[9]

#download iterating through each file name. This will all download to your␣
 ↪directory
base_url='https://www.ngdc.noaa.gov/thredds/fileServer/tiles/tiled_'+res+'//'␣
 ↪#base url for download
i=0
for i in range(0,len(link_df)):
    file_name=link_df['filename'][i]
    dwnld_link=base_url+file_name
    wget.download(dwnld_link,out = os.getcwd())
    i=i+1
```

```
100%
[…]
396236 / 396236
```

0.2.2 This cell converts all of the downloaded cudem nc's into .csv files with fields for latitude, longitude, depth, and crs. This may take a bit, especially with larger files.

[30]:
```
#Convert downloaded nc files to csv


for filename in os.listdir():
    if filename.endswith(".nc"):  #find the .nc files in your directory
        try:
            nc = netCDF4.Dataset(os.path.join(os.getcwd(), filename), mode='r')
            #establish naming components
            file_name_no_ext=os. path. splitext(filename)[0]
            out_name=os.getcwd()+'/'+file_name_no_ext+'.csv'
            #create a pandas dataframe
            df = pd.DataFrame(columns = ['latitude','longitude','depth','crs'])
            #assign values from nc files to dataframe
            df['latitude']=nc.variables['lat'][:]
            df['longitude']= nc.variables['lon'][:]
            df['depth']=nc.variables['Band1'][:]
            df['crs']=nc.variables['crs'][:]
            df.to_csv(out_name)
        except:
            pass
    else:
        continue
print('Data Conversion Sucessful!')
```

```
Data Conversion Sucessful!
```

This cell converts combines all of the csv's into one dataframe

[31]:
```
#merge csv's into one df
os.chdir(os.getcwd())

#create list of files in folder
all_filenames = [i for i in glob.glob('*.csv')]
#combine all files in the list
combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])

cudem_df=combined_csv
```

let's take a look at that combined file

[32]:
```
cudem_df
```

[32]:

	Unnamed: 0	latitude	longitude	depth	crs
0	0	39.245417	-71.754583	-2101.8882	NaN
1	1	39.246250	-71.753750	-2100.2825	NaN
2	2	39.247083	-71.752917	-2097.6938	NaN
3	3	39.247917	-71.752083	-2095.3445	NaN
4	4	39.248750	-71.751250	-2094.5290	NaN
..	…	…	…	…	…
307	307	38.251250	-70.998750	-2989.7942	NaN
308	308	38.252083	-70.997917	-2989.3335	NaN
309	309	38.252917	-70.997083	-2989.2550	NaN
310	310	38.253750	-70.996250	-2988.9983	NaN
311	311	38.254583	-70.995417	-2988.4100	NaN

[33384 rows x 5 columns]

This cell will delete all of the uncombined csv's and raw CUDEM files to clean up your folder

[33]:
```
for filename in os.listdir(os.getcwd()):
    if filename.startswith("ncei"):  #find the ncei cudem files and csv's in␣
 ↪your folder
        try:
            #establish naming components
            out_name=os.getcwd()+'/'+filename
            os.remove(out_name)

        except:
            pass
    else:
        continue
print("Source files deleted")
```

```
Source files deleted
```

Download the sample dataset

[34]:
```
url = 'https://zenodo.org/record/6099266/files/'
if zen.value=='Entire Dataset':
    filename='dataset_10kmcoast.csv'
if zen.value=='Estimated Onshore Data':
    filename='Data_EstimatedOnshore.csv'
# if zen.value=='Verified Onshore Data':
#     filename='Data_VerifiedOnshore.csv'
# if zen.value=='Verified Onshore Post 2012 Data':
#     filename='Data_Post2012_VerifiedOnshore.csv'
print("Downloading {}".format(url+filename))
```

```
Downloading https://zenodo.org/record/6099266/files/dataset_10kmcoast.csv
```

The next cell will download the CGRASP dataset and read it in as a pandas dataframe with

variable name `sample_df`

[35]:
```
url=(url+filename)
print('Retrieving Data, Please Wait')
#retrieve data
sample_df=pd.read_csv(url)
print('Sediment Data Retrieved!')
```

```
Retrieving Data, Please Wait
Sediment Data Retrieved!
```

```
/tmp/ipykernel_881750/3103848892.py:4: DtypeWarning: Columns (6,11) have mixed
types. Specify dtype option on import or set low_memory=False.
  sample_df=pd.read_csv(url)
```

Let's take a quick look at the top of the file

[36]:
```
sample_df.head()
```

[36]:

	ID	Sample_ID	Sample_Type_Code	Project	dataset	Date	\
0	81	NaN	NaN	ussb_project_259	US_SeaBed	NaN	
1	80	NaN	NaN	ussb_project_259	US_SeaBed	NaN	
2	85	NaN	NaN	ussb_project_115	US_SeaBed	NaN	
3	86	NaN	NaN	ussb_project_115	US_SeaBed	NaN	
4	88	NaN	NaN	ussb_project_115	US_SeaBed	NaN	

	Location_Type	latitude	longitude	Contact	…	d16	d25	d30	d50	d65	d75	\
0	NaN	25.96090	-97.12251	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	25.96090	-97.12251	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	25.96667	-97.08334	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	25.96667	-97.08334	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	25.96667	-97.09972	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	

	d84	d90	d95	Notes
0	NaN	NaN	NaN	LocnName: Chart_11301_2001_3;ObsvnKey: 302126;…
1	NaN	NaN	NaN	LocnName: Chart_11301_2001_3;ObsvnKey: 302126;…
2	NaN	NaN	NaN	LocnName: HE-20-2-92: H-10429: (CatNo: 48734);…
3	NaN	NaN	NaN	LocnName: HE-20-2-92: H-10429: (CatNo: 48734);…
4	NaN	NaN	NaN	LocnName: HE-20-2-92: H-10429: (CatNo: 48735);…

[5 rows x 34 columns]

0.3 Now lets make use of both datasets to assign depths from CUDEM to C-Grasp Samples

Turn the sample and CUDEM datasets in to GeoDataFrames (spatial data) and set them to their CRS(EPSG:4326) and then project them to a projected coordinate system (EPSG 3857)

[37]:
```
#convert the CUDEM dataframe

cudem_gdf = gpd.GeoDataFrame(
    cudem_df, geometry=gpd.points_from_xy(cudem_df.longitude, cudem_df.
 ↪latitude))

cudem_gdf=cudem_gdf.set_crs('EPSG:4326')
cudem_gdf=cudem_gdf.to_crs('EPSG:3857')


#convert the C-GRASP dataframe
sample_gdf = gpd.GeoDataFrame(
    sample_df, geometry=gpd.points_from_xy(sample_df.longitude, sample_df.
 ↪latitude))

sample_gdf=sample_gdf.set_crs('EPSG:4326')
sample_gdf=sample_gdf.to_crs('EPSG:3857')
```

This cell will use the package Geopandas' sjoin_nearest function to join together the sediment samples with its nearest CUDEM depth measurement.

[38]:
```
joined = sample_gdf.sjoin_nearest(cudem_gdf, how="left")

joined=joined.to_crs('EPSG:4326') #convert back to epsg 4326

df=pd.DataFrame(joined)
```

Rename the fields in the dataframe to be more recognizeable (i.e. turn "left" and "right" fields to "sample" and "cudem"

[39]:
```
#rename some of the new field names
df['latitude_sample']=df['latitude_left']
df['latitude_cudem']=df['latitude_right']
df['longitude_sample']=df['longitude_left']
df['longitude_cudem']=df['longitude_right']

#drop geometry column
df=df.
 ↪drop(columns=['geometry','crs','latitude_left','latitude_right','longitude_left','longitude
```

Let's take a look to see how that all worked out:

[40]:
```
df
```

[40]:

	ID	Sample_ID	Sample_Type_Code	Project	\
0	81	NaN	NaN	ussb_project_259	
1	80	NaN	NaN	ussb_project_259	
2	85	NaN	NaN	ussb_project_115	

```
3           86       NaN               NaN  ussb_project_115
4           88       NaN               NaN  ussb_project_115
…          …       …                 …              …
435196  636550       NaN               NaN  ussb_project_251
435197  636549       NaN               NaN  ussb_project_251
435198  636554      H152               NaN         USGS/WHOI
435199  636553      H151               NaN         USGS/WHOI
435200  636555      H153               NaN         USGS/WHOI

                                          dataset        Date Location_Type  \
0                                        US_SeaBed         NaN           NaN
1                                        US_SeaBed         NaN           NaN
2                                        US_SeaBed         NaN           NaN
3                                        US_SeaBed         NaN           NaN
4                                        US_SeaBed         NaN           NaN
…                                              …         …           …
435196                                   US_SeaBed         NaN           NaN
435197                                   US_SeaBed         NaN           NaN
435198  USGS East Coast Sediment Texture Database  1964-11-06           NaN
435199  USGS East Coast Sediment Texture Database  1964-11-06           NaN
435200  USGS East Coast Sediment Texture Database  1964-11-06           NaN

       Contact  num_orig_dists Measured_Distributions  …  d90  d95  \
0          NaN               0                    NaN  …  NaN  NaN
1          NaN               0                    NaN  …  NaN  NaN
2          NaN               0                    NaN  …  NaN  NaN
3          NaN               0                    NaN  …  NaN  NaN
4          NaN               0                    NaN  …  NaN  NaN
…         …              …                    …  …  …  …
435196     NaN               0                    NaN  …  NaN  NaN
435197     NaN               0                    NaN  …  NaN  NaN
435198     NaN               0                    NaN  …  NaN  NaN
435199     NaN               0                    NaN  …  NaN  NaN
435200     NaN               0                    NaN  …  NaN  NaN

                                                  Notes index_right  \
0       LocnName: Chart_11301_2001_3;ObsvnKey: 302126;…           0
1       LocnName: Chart_11301_2001_3;ObsvnKey: 302126;…           0
2       LocnName: HE-20-2-92: H-10429: (CatNo: 48734);…           0
3       LocnName: HE-20-2-92: H-10429: (CatNo: 48734);…           0
4       LocnName: HE-20-2-92: H-10429: (CatNo: 48735);…           0
…                                                     …         …
435196  LocnName: H010;ObsvnKey: 214472;Device: UnidDe…         311
435197  LocnName: H010;ObsvnKey: 214472;Device: UnidDe…         311
435198  Listed Location: WOODSTOCK,N.BRUNSWICK Litholo…         311
435199  Listed Location: WOODSTOCK,N.BRUNSWICK Litholo…         311
435200  Listed Location: WOODSTOCK,N.BRUNSWICK Litholo…         311
```

```
        Unnamed: 0      depth  latitude_sample latitude_cudem  \
0                0 -1705.9852         25.96090      37.745417
1                0 -1705.9852         25.96090      37.745417
2                0 -1705.9852         25.96667      37.745417
3                0 -1705.9852         25.96667      37.745417
4                0 -1705.9852         25.96667      37.745417
…              …         …               …             …
435196         311 -1362.2013         45.10000      39.754583
435197         311 -1362.2013         45.10000      39.754583
435198         311 -1362.2013         46.08667      39.754583
435199         311 -1362.2013         46.08667      39.754583
435200         311 -1362.2013         46.08667      39.754583

        longitude_sample longitude_cudem
0              -97.12251      -73.754583
1              -97.12251      -73.754583
2              -97.08334      -73.754583
3              -97.08334      -73.754583
4              -97.09972      -73.754583
…                    …              …
435196         -67.63167      -70.995417
435197         -67.63167      -70.995417
435198         -67.55000      -70.995417
435199         -67.55000      -70.995417
435200         -67.55000      -70.995417

[435408 rows x 39 columns]
```

0.3.1 Write to file

Finally, define a csv file name for the output dataframe

[ ]:
```
output_csvfile='../data_CudemDepths.csv'
```

write the data to that csv file

[ ]:
```
df.to_csv(output_csvfile)
```