```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
This script is used to download publicly available topography data
(NASADEM or Tandem-X)
Author: Alexandra Christensen
Created: Wed Oct 14 10:25:58 2020
Updated: 12/01/2022

"""

import rasterio
from osgeo import ogr
from osgeo import gdal
import numpy as np
import geopandas as gpd
import zipfile
import certifi
import urllib3
from zipfile import ZipFile
#http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED',
ca_certs=certifi.where())
import requests
import pandas as pd
import math
import os
import fnmatch
import time

def
download_NASADEM2(source,delta,nasadem_path,folders,x1,x2,y1,y2,zone,ulx,
lry,lrx,uly,xres,yres):
    ######################
    ## Use GEBCO extents from Dewi to deteremine model extents
    #src =
gdal.Open('/users/alchrist/documents/externaldata/deltas_gebco/%s.tif'
%(delta))
    #ulx,xres,xskew,uly,ysken,yres = src.GetGeoTransform()
    #x1 = math.floor(ulx)
    #y1 = math.floor(uly)
    #x2 = math.floor(lrx)
    #y2 = math.floor(lry)
    #x2 = math.floor(ulx+src.RasterXSize*xres)
    #y2 = math.floor(uly+src.RasterYSize*yres)
    #zone = int(np.ceil((ulx + 180)/6))
    ytiles = []
    ytiles2 = []
    xtiles = []
    xtiles2 = []
    #print(y1,y2,x1,x2)
    if y1>=0 and y2>=0:
        NS = 'n'
        EPSG = 32200+zone
```

```
        #ytiles = range(int(y1)-
math.ceil(src.RasterYSize*abs(yres)),y1+1,1)
        y1 = y1+1 #extend extent
        y2 = y2-1 #extend extent
        ytiles = range(y2,y1+1,1)
    elif y1>=0 and y2<0:
        NS = 'n'
        NS2 = 's'
        EPSG = 32200+zone
        y = abs(y2)+1
        y1 = y1+1 #extend extent
        ytiles = range(0,y1+1,1)
        ytiles2 = range(0,y+1+1,1)
    else:
        NS = 's'
        y = abs(y1)
        EPSG = 32700+zone
        #ytiles =
range(y,int(y)+math.ceil(src.RasterYSize*abs(yres))+1,1)
        y1 = abs(y1)-1 #extend extent
        y2 = abs(y2)+1 #extend extent
        ytiles = range(abs(y1),abs(y2)+1,1)
    if x1>=0:
        EW = 'e'
        #xtiles =
range(x1,int(x1)+math.ceil(src.RasterXSize*abs(xres))+1,1)
        x1 = x1-1 #extend extent
        x2 = x2+1
        xtiles = range(x1,x2+1,1)
    elif x1<0 and x2>=0:
        EW = 'e'
        EW2 = 'w'
        x = abs(x2)+1
        x1 = x1+1 #extend extent
        xtiles = range(0,x1+1,1)
        xtiles2 = range(0,x+1+1,1)
    else:
        EW = 'w'
        x = abs(x1)
        x1 = x1-1 #extend extent
        x2 = x2+1 #extend extent
        #xtiles = xrange(int(x)-math.ceil(src.RasterXSize*xres)-1,x,1)
        xtiles = range(abs(x2),abs(x1)+1,1)
    #print('UTM Zone: %s%s EPSG: %s' %(zone,NS,EPSG))
    ########################
    # Download NASADEM tiles
    nasadems = []
    reqs = 0

    #print(xtiles,xtiles2, ytiles, ytiles2)
    for xtile in xtiles:
        print('X = %s' %(xtile))
        for ytile in ytiles:
            print('North Y = %s' %(ytile))
```

```
            for attempt in range(0,10):
                try:
                    if source=='NASADEM':
                        url =
'https://e4ftl01.cr.usgs.gov/MEASURES/NASADEM_HGT.001/2000.02.11/NASADEM_
HGT_%s%s%s%s.zip' %(NS,str(ytile).zfill(2),EW,str(xtile).zfill(3))
                        filetype = 'zip'
                    elif source == 'GLO30':
                        url = 'https://copernicus-dem-
30m.s3.amazonaws.com/Copernicus_DSM_COG_10_%s%s_00_%s%s_00_DEM/Copernicus
_DSM_COG_10_%s%s_00_%s%s_00_DEM.tif'
%(NS.capitalize(),str(ytile).zfill(2),EW.capitalize(),str(xtile).zfill(3)
,NS.capitalize(),str(ytile).zfill(2),EW.capitalize(),str(xtile).zfill(3))
                        filetype = 'tif'
                    result = requests.get(url)
                    result.raise_for_status()
                    f =
open(nasadem_path+folders[0]+source+'_%s%s%s%s.%s'
%(NS,str(ytile).zfill(2),EW,str(xtile).zfill(3),filetype),'wb')
                    f.write(result.content)
                    f.close()
                    print('contents of URL written to %s_%s%s%s%s.%s'
%(source,NS,str(ytile).zfill(2),EW,str(xtile).zfill(3),filetype))
                    if filetype == 'zip':
                        try:
                            with
ZipFile(nasadem_path+folders[0]+source+'_%s%s%s%s.zip'
%(NS,str(ytile).zfill(2),EW,str(xtile).zfill(3)),'r') as zp:
                                zp.extractall(nasadem_path + folders[0])
                            nasadems.append(nasadem_path +folders[0] +
'%s%s%s%s.hgt' %(NS,str(ytile).zfill(2),EW,str(xtile).zfill(3)))
                        except:
                            ''# print('out of bounds')
                    elif filetype == 'tif':
                         nasadems.append(nasadem_path + folders[0] +
source+ '_%s%s%s%s.tif' %(NS,str(ytile).zfill(2),EW,str(xtile).zfill(3)))
                except requests.exceptions.ConnectionError:
                    print('Connection error - waiting 60 seconds')
                    time.sleep(60)
                except requests.exceptions.RequestException:
                    #print('requests.get() returned an error code
'+str(result.status_code))
                    if attempt == 9: print('Tried to connect 10 times and
failed - moving to the next tile. Probably no tile here (ocean?)')
                else:
                    break

        for ytile2 in ytiles2:
            print('South Y = %s' %(ytile2))
            for attempt in range(0,10):
                try:
                    if source=='NASADEM':
```

```
                        url =
'https://e4ftl01.cr.usgs.gov/MEASURES/NASADEM_HGT.001/2000.02.11/NASADEM_
HGT_%s%s%s%s.zip' %(NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3))
                        filetype = 'zip'
                    elif source == 'GLO30':
                        url = 'https://copernicus-dem-
30m.s3.amazonaws.com/Copernicus_DSM_COG_10_%s%s_00_%s%s_00_DEM/Copernicus
_DSM_COG_10_%s%s_00_%s%s_00_DEM.tif'
%(NS2.upper(),str(ytile2).zfill(2),EW.upper(),str(xtile).zfill(3),NS2.upp
er(),str(ytile2).zfill(2),EW.upper(),str(xtile).zfill(3))
                        filetype = 'tif'
                    result = requests.get(url)
                    result.raise_for_status()
                    f = open(nasadem_path +
folders[0]+source+'_%s%s%s%s.%s'
%(NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3),filetype),'wb')
                    f.write(result.content)
                    f.close()
                    print('contents of URL written to %s_%s%s%s%s.%s'
%(source,NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3),filetype))
                    if filetype == 'zip':
                        try:
                            with ZipFile(nasadem_path +
folders[0]+source+'_%s%s%s%s.zip'
%(NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3)),'r') as zp:
                                zp.extractall(nasadem_path + folders[0])
                            nasadems.append(nasadem_path + folders[0] +
'%s%s%s%s.hgt' %(NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3)))
                        except:
                            ''# print('out of bounds')
                    elif filetype == 'tif':
                         nasadems.append(nasadem_path + folders[0] +
source+ '_%s%s%s%s.tif'
%(NS2,str(ytile2).zfill(2),EW,str(xtile).zfill(3)))
                except requests.exceptions.ConnectionError:
                    print('Connection error - waiting 60 seconds')
                    time.sleep(60)
                except requests.exceptions.RequestException:
                    #print('requests.get() returned an error code
'+str(result.status_code))
                    if attempt ==9:print('Tried to connect 10 times and
failed - moving to the next tile')
                else:
                    break
    with open("%s%s_%s.txt" %(nasadem_path + folders[0],delta,source),
'w') as f:
        for item in nasadems:
            f.write("%s\n" % item)
    # Merge NASADEM tiles to make topography file
    os.system("gdalbuildvrt %s%s_%s.vrt -input_file_list %s%s_%s.txt -
vrtnodata -9999 -a_srs EPSG:4326+5733" %(nasadem_path +
folders[0],delta,source,nasadem_path + folders[0],delta,source))
    # for t in nasadems:
    delete = [os.path.join(dirpath,f)
```

```
        for dirpath,dirnames, files in os.walk(nasadem_path + folders[0])
        for f in fnmatch.filter(files,'*.zip')]
    for d in delete:
        os.remove(d)

def other(x):
    warped = gdal.Warp('', src, dstSRS='EPSG:%s' %(EPSG),
format='VRT',xRes=resolution,yRes=resolution)
    ulx, xres, xskew, uly, ysken, yres = warped.GetGeoTransform()
    extentarea = abs(warped.RasterXSize*xres*warped.RasterYSize*yres)
    print('Approximate area of model domain is %s km^2'
%(extentarea/1000000))
    if (extentarea/1000000)> 100000:
        cutby = 30*600
    else:
        cutby = 30*60
    print('Cutby = %s' %(cutby))
    ulx = ulx + cutby
    uly = uly - cutby
    lrx = ulx + (warped.RasterXSize * xres) - (cutby*2)
    lry = uly + (warped.RasterYSize * yres) + (cutby*2)
    NASADEMvrt = gdal.Open(deltapath + '%s_NASADEM2.tif' %(delta))
    NASADEM = gdal.Warp(deltapath + '%s_NASADEM.tif' %(delta),
NASADEMvrt, dstSRS='EPSG:%s' %(EPSG), format='GTiff',xRes =30, yRes=30,
targetAlignedPixels = True, outputBounds = [ulx,lry,lrx,uly])
```