Subbasin

March 10, 2021

1 This script intends to create the subbasin map of a wtershed using the Physitel inputs/outputs. The script heavily relies on arcpy library for geospatial calculations and runs in Python version 2.7.

1.0.1 section1: importing the libraries

```
In [1]: import arcpy,os,re
        from arcpy.sa import *
        from arcpy import env
        import pandas as pd
        import scipy.io as sio
        import shutil
```

1.0.2 section2: reading the river reach data from the .mat file

```
In [2]: # as the reach data will be read from a MATLAB database (provided by DEH), we should f
        Troncon_path = r'C:\Users\mohbiz1\Desktop\Dossier_travail\Hydrotel\DEH\INFO_TRONCON.ma
        data = sio.loadmat(Troncon_path, struct_as_record=False, squeeze_me=True)
        region_name = data['SLNO_TRONCON']
        size = region_name.shape[0]
        s = size-1  # size = number of river reaches

In [3]: # reading the attributes associated with each river reach in data such as length, widt

        df = []
        for i in range(size):
            rec = region_name[i]
            df.append([rec.NOEUD_AVAL.NUMERO,rec.NOEUD_AMONT.NUMERO,rec.NO_TRONCON,rec.TYPE_NO
        TRONCON_INFO= pd.DataFrame(df,columns = ['NODE_AVAL','NODE_AMONT','SubId','TYPE_NO','R

In [4]: # here we define the directory in whcih the results will be written.
        pathtoDirectory = r"C:\Users\mohbiz1\Desktop\Dossier_travail\Hydrotel\DEH\MG24HA\SLNO_
        workspace = os.path.join(pathtoDirectory+ "\HRU")
        shutil.copytree(pathtoDirectory,workspace)
        # Set environment workspace for arcpy
        arcpy.env.overwriteOutput = True
        env.workspace = r"C:\Users\mohbiz1\Desktop\Dossier_travail\Hydrotel\DEH\MG24HA\SLNO_MG2
        workspace = arcpy.env.workspace # the geospatial calculations will be written in in th
```

```
In [5]: # add subbasin ID to the attribute table of uhrh map (created by physitel)
        arcpy.AddField_management("uhrh.shp", "SubId", "LONG", "", "", 16)

Out[5]: <Result 'uhrh.shp'>
```

```
In [ ]: # for lakes/reservoirs, the manning coefficient and bankfulldepth are null. They are r
        TRONCON_INFO.loc[TRONCON_INFO.TYPE_NO == 2, 'Ch_n'] = 0.
        TRONCON_INFO.loc[TRONCON_INFO.TYPE_NO == 2, 'BnkfWidth'] = 0.
```

```
In [ ]: # finding the subbasins associated with each river reach and write them to the SubId f
        for i in range(size):
            a = TRONCON_INFO['ASSOCI_UHRH'][i]
            id = TRONCON_INFO['SubId'][i]
            #print ("writing subbasin :", i )
            if type(a) is int:
                aa = [a]
                st = len(aa)
                stt = st-1
                dict = {i: aa[i] for i in range(0, len(aa))}
            else:
                al = a.tolist()
                st = len(al)  # number of UHRH associated with current reach
                stt = st - 1
                #create a temporary dictionary
                dict = {i: al[i] for i in range(0, len(al))}

            for j in range(st):
                with arcpy.da.UpdateCursor("uhrh.shp", ['SHAPE@', 'SubId','ident']) as rows:
                    for row in rows:
                        if row[2] in dict.values():
                            row[1] = id
                        rows.updateRow(row)
```

```
In [ ]: # merge the uhrhs based on SubId field. the number of feature classes in the resulting

        arcpy.MakeFeatureLayer_management("uhrh.shp","templayer") #create a temporary feature
        arcpy.Dissolve_management("templayer","uhrh_diss.shp","SubId","","","")
        arcpy.AddGeometryAttributes_management("uhrh_diss.shp", "AREA", "METERS", "SQUARE_KILO
```

```
In [ ]: # finding the downstream subbasin ID associated with each uhrh

        TRONCON_INFO['DownSubId']=-1
        for i in range(size):
            naval = TRONCON_INFO['NODE_AVAL'][i]
            for j in range(size):
                namont= TRONCON_INFO['NODE_AMONT'][j]
                id = TRONCON_INFO['SubId'][j]
                if type(namont) is int:
                    nal = [namont]
                else:
                    nal = namont.tolist()
                if naval in nal: # if naval (downstream node) for reach i is upstream node for
                    TRONCON_INFO.loc[i, 'DownSubId'] = id
```

```
In [ ]: # specify the lake subbasins. the BnkfDepth value, and a column representing whether t
        TRONCON_INFO['IsObs'] = (TRONCON_INFO['DownSubId'] == -1).astype(int)  #create a boole
        TRONCON_INFO['BnkfDepth'] = 0.13 * (TRONCON_INFO['SA_Up'] ** 0.4) # taken from equatio
        TRONCON_INFO['IsLake']=-9999.99
        TRONCON_INFO.loc[TRONCON_INFO.TYPE_NO == 2, 'IsLake'] = 1
```

```
In [ ]: # add the downstream ID as well as other required fields to the shapefile of the creat
        arcpy.AddField_management("uhrh_diss.shp", "DownSubId", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "Rivlen", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "BkfWidth", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "BkfDepth", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "IsObs", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "RivSlope", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "Ch_n", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "FloodP_n", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "IsLake", "Double", "", "", 16)
        arcpy.AddField_management("uhrh_diss.shp", "Type", "SHORT", "", "", 16) # NOT NEEDED f
        arcpy.AddField_management("uhrh_diss.shp", "HyLakeId", "Double", "", "", 16) # NOT NEE
```

```
In [ ]: #loop through feature classes and copy the datframe values to the attribute table of t
        j = 0
        t = 0
        with arcpy.da.UpdateCursor("uhrh_diss.shp", ("Rivlen", "DownSubId","SubId","IsObs",'Bk
             for ROW in cursor:
                  #ROW[0] = TRONCON_INFO["TYPE_NO"][j]
                  ROW[0] = TRONCON_INFO["Rivlen"][j]
                  ROW[1] = TRONCON_INFO["DownSubId"][j]
                  ROW[2] = TRONCON_INFO["SubId"][j]
                  ROW[3] = TRONCON_INFO["IsObs"][j]
                  ROW[4] = TRONCON_INFO["BnkfWidth"][j]
                  ROW[5] = TRONCON_INFO["BnkfDepth"][j]
                  ROW[6] = TRONCON_INFO["RivSlope"][j]
                  ROW[7] = TRONCON_INFO["Ch_n"][j]
                  ROW[8] = 0.1
                  ROW[9] = TRONCON_INFO["IsLake"][j]
                  if row[9]==1:
                      t = t+1
                      ROW[10] = t
                  cursor.updateRow(ROW)
                  j += 1

        del cursor
```