CODIGO PYTHON HEATMAP\_U

import processing

import os

#defineix el directori

folder="C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/"

#captura els arxius en una llista

filelist=os.listdir(folder)

feedback = QgsProcessingFeedback()

#recorre cada arxiu i si és un shape, executa el buffer.

for file in filelist:

if file.endswith('.shp'):

layer=QgsVectorLayer(folder+file,file,'ogr')

file2 = file.replace(".shp","")

output=str(folder + "heatmap/hm\_" + file2 + ".tif")

parameters={'INPUT':layer,'RADIUS':5000, 'PIXEL\_SIZE':500, 'OUTPUT':output}

processing.runAndLoadResults('qgis:heatmapkerneldensityestimation',parameters,feedback=feedback)

CODIGO PYTHON CUT\_BY\_MASK\_U

import processing

import os

#defineix el directori

folder="C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/heatmap/"

#captura els arxius en una llista

filelist=os.listdir(folder)

mask='C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/shp\_interviste\_Muga\_26\_nuovi/comarca\_alt\_emporda.shp'

feedback = QgsProcessingFeedback()

#recorre cada arxiu i si és un shape, executa el buffer.

for file in filelist:

if file.endswith('.tif'):

layer=QgsRasterLayer(folder+file,file)

folder2="C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/mask/"

output=str(folder2 + file)

parameters={'INPUT':layer, 'MASK':mask, 'NODATA':0, 'CROP\_TO\_CUTLINE':True, 'KEEP\_RESOLUTION': True, 'OUTPUT':output}

processing.run('gdal:cliprasterbymasklayer',parameters,feedback=feedback)

CODIGO PYTHON R\_SERIES\_U

import os

import shutil

#import lib.config as config

import processing

from osgeo import gdal

# Define variables

ROOT\_FOLDER = "C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/"

LAYER\_FOLDER = ROOT\_FOLDER + os.sep + 'mask' + os.sep

OUTPUT\_FOLDER = ROOT\_FOLDER + os.sep + 'resultados' + os.sep

filelist=os.listdir(LAYER\_FOLDER)

#uri\_comarca = '/home/josep/SIGTE/Enrica/base/comarca\_alt\_emporda.tif'

#comarca = 'comarca\_alt\_emporda@1'

uri\_comarca\_999 = "C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/comarca\_999.tif"

comarca\_999 = 'comarca\_999@1'

for file in filelist:

if file.endswith('.tif'):

uri = LAYER\_FOLDER + str(file)

nom = str(file.split(".")[0])+'@1'

sintax = { 'DATA\_TYPE' : 5, 'INPUT' : [uri\_comarca\_999,uri], 'NODATA\_INPUT' : None, 'NODATA\_OUTPUT' : None, 'OPTIONS' : '', 'OUTPUT' : OUTPUT\_FOLDER + file, 'PCT' : False, 'SEPARATE' : False }

processing.runAndLoadResults("gdal:merge", sintax)

CODIGO REPROJECT\_U

import processing

import os

#defineix el directori, ha d'acabar amb "/"

folder="C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/"

#captura els arxius en una llista

filelist=os.listdir(folder)

feedback = QgsProcessingFeedback()

#recorre cada arxiu i si és un shape, executa el buffer.

for file in filelist:

if file.endswith('.shp'):

print(file)

layer=QgsVectorLayer(folder+file,file,'ogr')

layer\_dp=layer.dataProvider()

#METODO crs() SOBRE EL DATA PROVIDER

layer\_crs=layer\_dp.crs()

#TRANSFORMA EL CRS EN UN STRING

layer\_crs\_str=layer\_crs.authid()

print ("crs d'entrada de "+ file+" és "+layer\_crs\_str)

#si creem una carpeta de destinació, ha d'existir, python no la crea i dona error

output=str(folder + "reproject/" + file)

parameters={'INPUT':layer, 'TARGET\_CRS':'EPSG:25831', 'OUTPUT':output}

processing.run('native:reprojectlayer',parameters,feedback=feedback)

print(output)

#ACCEDE A LA CAPA ACTIVA

layer2=QgsVectorLayer(output,file,'ogr')

#ACCEDE AL DATA PROVIDER DE LA CAPA

layer\_dp=layer2.dataProvider()

#METODO crs() SOBRE EL DATA PROVIDER

layer\_crs=layer\_dp.crs()

#TRANSFORMA EL CRS EN UN STRING

layer\_crs\_str=layer\_crs.authid()

print ("crs de sortida de "+ file+" és "+layer\_crs\_str)

CODIGO PYTHON RASTER CALCULATOR\_NU (Suma los raster de una carpeta y genera otro nuevo con la suma de todos)

import os

import fnmatch

import os

import shutil

#import lib.config as config

import processing

from osgeo import gdal

#from PyQt4.QtCore import QFileInfo

from qgis.analysis import QgsRasterCalculator, QgsRasterCalculatorEntry

inputFolder = "C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/hm\_vul"

outputfilename = "C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/calculos\_raster/hotspot\_vul\_py2.tif"

outputFolder="C:/++ELABORAZIONI\_NUOVE/entrevistas/+++HEATMAP/calculos\_raster"

def findRasters(path, filter):

for root, dirs, files in os.walk(path, filter):

for file in fnmatch.filter(files, filter):

yield os.path.join(root, file)

entries = []

layers = []

for l in findRasters(inputFolder, '\*.tif'):

print (l)

fileInfo = QFileInfo(l)

baseName = fileInfo.baseName()

rlayer = QgsRasterLayer(l,baseName)

#QgsMapLayerRegistry.instance().addMapLayer(rlayer)

layer = QgsRasterCalculatorEntry()

layer.ref = rlayer.name() + '@1'

layer.raster = rlayer

layers.append(rlayer)

layer.bandNumber = 1

print (layer)

entries.append(layer)

reflist = " + ".join([ent.ref for ent in entries])

expression = '(' + reflist + ')'

outuput = outputfilename

print (expression)

calc = QgsRasterCalculator(expression,outputfilename,'GTiff', layers[0].extent(), layers[0].width(), layers[0].height(), entries)

calc.processCalculation()