

# 01\_Retrieve\_network\_data

October 14, 2020

## 1 Extract street networks from OpenStreetMap

Computational notebook 01 for Climate adaptation plans in the context of coastal settlements: the case of Portugal.

Date: 27/06/2020

---

Input data contains manually digitised building footprints stored in geopackages divided according to geographical locations (following division proposed by Ribeiro (1945)).

Structure of GeoPackages:

```
./data/  
  atlantic.gpkg  
    name_blg - Polygon layers  
    name_blg  
    name_blg  
    ...  
  preatl.gpkg  
    name_blg  
    name_blg  
    name_blg  
    ...  
  premed.gpkg  
    name_blg  
    name_blg  
    name_blg  
    ...  
  med.gpkg  
    name_blg  
    name_blg  
    name_blg  
    ...
```

CRS of the original data is EPSG:3763.

```
<Projected CRS: EPSG:3763>  
Name: ETRS89 / Portugal TM06  
Axis Info [cartesian]:
```

- X[east]: Easting (metre)
- Y[north]: Northing (metre)

Area of Use:

- name: Portugal - mainland - onshore
- bounds: (-9.56, 36.95, -6.19, 42.16)

Coordinate Operation:

- name: Portugal TM06
- method: Transverse Mercator

Datum: European Terrestrial Reference System 1989

- Ellipsoid: GRS 1980
- Prime Meridian: Greenwich

This notebook downloads and clips street network within 2500m radius around input data convex hull. During the extraction it plots resulting layers for visual inspection.

```
[4]: import fiona
import geopandas as gpd
import osmnx as ox
import matplotlib
import matplotlib.pyplot as plt
```

```
[5]: fiona.__version__, gpd.__version__, ox.__version__, matplotlib.__version__
```

```
[5]: ('1.8.13', '0.7.0', '0.11.4', '3.2.1')
```

```
[2]: parts = ['atlantic', 'preatl', 'premed', 'med']
folder = 'data/'

for part in parts:
    path = folder + part + '.gpkg'
    layers = [x for x in fiona.listlayers(path) if 'blg' in x]

    for l in layers:
        print(l)
        blg = gpd.read_file(path, layer=l)
        union = gpd.GeoSeries(blg.buffer(0).unary_union.centroid, crs=blg.crs).
        ↳to_crs(epsg=4326).iloc[0]
        location_point = (union.y, union.x)

        streets_graph = ox.graph_from_point(location_point, distance=5000,
        ↳distance_type='bbox', network_type='drive')
        streets_graph = ox.project_graph(streets_graph)
        streets_graph = ox.get_undirected(streets_graph)

        edges = ox.save_load.graph_to_gdfs(streets_graph, nodes=False,
        ↳edges=True,
                                         node_geometry=False,
        ↳fill_edge_geometry=True)
```

```
edges = edges.to_crs(epsg=3763)

clip = blg.unary_union.convex_hull.buffer(2500)

clipped_edges = edges.intersection(clip)

clipped_edges = clipped_edges.loc[~clipped_edges.is_empty]

ax = clipped_edges.plot(linewidth=0.2, figsize=(16, 16))
blg.plot(ax=ax, color='r')

clipped_edges.to_file(path, layer=1[:-3] + 'str', driver='GPKG')
```