Generation of Multi-level Directed Graphs Arbitrary Geometry - Part 2

Boris Petelin

```
library(knitr)
```

```
knitr::opts_chunk$set(cache=FALSE)
```

```
import sys
import os
import fastkml
from shapely.geometry import Polygon
mldgpath = "C:/Users/petelin/Documents/MLDG"
sys.path.insert(0, mldgpath)
import mldg

model = 'mfs'
modelpath = mldgpath + "/models/" + model
configfile = modelpath + "/" + model + '.cfg'
maskfile = modelpath + '/' + model + '_mask.shp'
mask = mldg.load_mask(maskfile)

projspath = mldgpath + "/" + "projects"
projectsuffix = "seas_manual"
project = model + "_" + projectsuffix
projpath = projspath + '/' + project
geompath = projpath + '/geometry'

kmlfile = geompath + '/MED.kml'
doc = file(kmlfile).read()
k = fastkml.KML()
k.from_string(doc)
f1 = list(k.features())
f2 = list(f1[0].features())
f3 = list(f2[0].features())

sea_geom = []
sea_name = []
sea_long_name = []
for i in range(len(f3)):
    sea_geom.append(f3[i].geometry)
    sea_name.append(f3[i].name)
    sea_long_name.append(f3[i].name)

vertfile = geompath + '/' + model + '_vertices_' + projectsuffix + '.shp'
include_region = [2]
(verts,name,long_name,lon_wmc,lat_wmc) = \
mldg.generate_arbitrary_vertices(sea_geom, sea_name,sea_long_name,mask, \
                            include_region,configfile)
```

```
len1 = len(verts)
for i in range(len1):
    for j in range(i+1,len1):
        inters = verts[i].intersection(verts[j])
        if (not inters.is_empty):
            verts[i] = verts[i].difference(verts[j])

mldg.show_vertices_mask(verts,name,lon_wmc,lat_wmc,mask,configfile)
mldg.save_vertices(verts,name,long_name,lon_wmc,lat_wmc,vertfile)

vertfile_aggr = geompath + '/' + model + '_vertices_' + projectsuffix + \
'_aggregated.shp'

old_verts = [['ALB'],
             ['SWE'],
             ['NWE1','NWE2'],
             ['TYR1','TYR2'],
             ['ADR'],
             ['ION1','ION2','ION3'],
             ['CEN1','CEN2'],
             ['AEG'],
             ['NLE'],
             ['SLE']]

new_verts = ['ALB',
             'SWE',
             'NWE',
             'TYR',
             'ADR',
             'ION',
             'CEN',
             'AEG',
             'NLE',
             'SLE']

new_long_name = ['Alboran',
                 'South-Western',
                 'North-Western',
                 'Tyrrhenian',
                 'Adriatic',
                 'Ionian',
                 'Central',
                 'Aegean',
                 'North-Levantine',
                 'South-Levantine']


(verts_aggr,name_aggr,long_name_aggr,lon_wmc_aggr,lat_wmc_aggr) = \
mldg.aggregate_vertices(verts,name,old_verts,new_verts,new_long_name)

mldg.show_vertices_mask(verts_aggr,name_aggr,lon_wmc_aggr,lat_wmc_aggr,mask,configfile)

print vertfile_aggr
```

```
mldg.save_vertices(verts_aggr,name_aggr,long_name_aggr,lon_wmc_aggr,lat_wmc_aggr,vertfile_aggr)
```

```
## C:/Users/petelin/Documents/MLDG/projects/mfs_seas_manual/geometry/mfs_vertices_seas_manual_aggregate
## No handlers could be found for logger "fastkml.config"
```