French Forest Sector Model (FFSM++)

Laboratoire d'Economie Forestière - LEF - Nancy, France

Preparation of the spatial data (input GIS layers)

This page highlight the technical data of the reference system used in the current FFSM version and indicates how to reproduce them, e.g. for countries other than France or for update the Land Cover data. Data manipulation can be done using the open source software Quantum Gis (Quantum GIS Development Team, 2012) and Grass (GRASS Development Team, 2012). The next table shows the reference system of the spatial data. Their projection match the one of the CLC06 as distributed from the European Environmental Agency.

Parameter	Value
EPSG ID	3035
projection	99 (Lambert Azimuthal Equal Area)
zone	0
datum	etrs89
ellipsoid	grs80
north	3168000
south	1992000
west	2976000
east	4512000
nsres	8000
ewres	8000
rows	147
cols	192
cells	28224

Common operations required to build the datasets

Change a map projection

To reproject a map (e.g. NUTS) :

- Open a map with the target projection in Qgis;
- Right click the theme and from the drop-down menu select “use project projection based on this reference system”;
- Open the shapefile you want to convert;
- In the project properties select “on the fly projection”;
- Save the second shapefile “save as shapefile” and change the projection field option in “project projections”.

Clip a shapefile

In order to e.g. clip the CLC06 shapefile only on France borders, the command in QGis is:
Vector → Geo-processing tools → Intersection

Map dissolve

In order to dissolve a shapefile (e.g. to dissolve selected nuts2 for metro France in a single nuts1 map):
Vector → Geo-processing tools → Dissolve
To add to the newly created map the correct area column (in square meters):
Vector → Geometry tools → Add or delete geometric columns

DTM (Digital Terrain Model)

The DTM used in the model derives from the official IGN BD ALTI (modèles numériques de terrain - MNT) downloadable from IGN [http://professionnels.ign.fr/bdalti#tab-3].
The option chosen is the 1000m resolution in format ASC (Télécharger la BD ALTI® 1000 m Métropole (format ASC) (zip, 2,1 Mo)).
The problem is that it comes in RGF93/Lambert-93 projection system (EPSG:2154). To convert it in EPSG:3035 one has to create a GRASS location with the EPSG:2154 projection (source); import the raster data (ESRI ascii format); create an other location with the EPSG:3035 (destination); import in destination the administrative borders; convert the admin borders from vector to raster; use the admin border raster as mark (r.mask); and finally use r.proj to import and reproject from source to dest the dtm file.
Finally the imported dtm can be exported as ascii grid file.

An alternative very precise DTM dataset for the whole Europe can be downloaded from the EEA web site [http://www.eea.europa.eu/data-and-maps/data/eu-dem#tab-european-data].

To fill eventual null values run from GRASS:

```
r.fillnulls input=dtm output=dtm_nonulls
```

In order to compute the mortality parameters based on regions and elevation run in grass:

```
r.mapcalc 'mortality_multiplier200 = if( (regions == 61 || regions == 62 || regions == 81 || regions == 82 || regions == 83 || regions == 71 ) && (dtm_nonulls <= 500), 2.0, 1.0)'
```

Maps rasterisation and Corine Land Cover map

Start GRASS, choose a folder where to keep your data, and click the Location Wizard to create a new Location. Give it a name and select as method to create the location to specify the reference systems from an EPSG code (e.g. 3035).
To work with the default France data, set the regions details as in Table 20, for all other cases do not set the default region extend and resolution at this time (you'll do later after having imported a shapefile).
Import a vector shapefile having the correct projection and area boundaries (e.g. the nuts of the country under interest).

Set the region to match the newly imported vector map, setting the desired resolution (e.g. 8000) and clicking the option to align the region to resolution.
Corine Land Cover maps, in vector format for a specific land use category, can be downloaded from the EAAE website [http://www.eea.europa.eu/data-and-maps/data/clc-2006-vector-data-version-3].
In order to create the rasters of 8 km with inside the values of the area covered by the clc06_{311|312|313} layers run the following script:

```
# First set resolution to 20 m..
g.region -p res=20

# Now rasterize the vector. This will give you as raster of 20m x 20m with values of 400 where the original vector covered, and null otherwise:
v.to.rast clc06_311_v out=clc06_311_r20m type=area use=val val=400
v.to.rast clc06_312_v out=clc06_312_r20m type=area use=val val=400
v.to.rast clc06_313_v out=clc06_313_r20m type=area use=val val=400

# Now change back the resolution to 8km..
g.region -p res=8000

# Use the "sum" method of r.resamp.stats to get the values for the new, coarse raster. This will leave you with a new raster of res 8km and with values 0-64000000:
r.resamp.stats clc06_311_r20m out=clc06_311_8km_sqm method=sum
r.resamp.stats clc06_312_r20m out=clc06_312_8km_sqm method=sum
r.resamp.stats clc06_313_r20m out=clc06_313_8km_sqm method=sum
```

Availability coefficient

Use the above script (within GRASS) to produce an availability coefficient based on the presence of protected areas and altimetry.
The shapefile of Nationally designated areas (CDDA) by IUCN category can be retrieved from the EEA website [http://www.eea.europa.eu/data-and-maps/data/nationally-designated-areas-national-cdda-9] (we used version 12 of 2014)

```
## Protected areas by IUCN categories ##

# Use he same method of CLC to compute the area in the pixel of protected areas for each IUCN category
g.region -p res=20
v.to.rast cdda_1a_v out=cdda_1a_r20m type=area use=val val=400
v.to.rast cdda_1b_v out=cdda_1b_r20m type=area use=val val=400
v.to.rast cdda_2_v out=cdda_2_r20m type=area use=val val=400
v.to.rast cdda_3_v out=cdda_3_r20m type=area use=val val=400
v.to.rast cdda_4_v out=cdda_4_r20m type=area use=val val=400
v.to.rast cdda_5_v out=cdda_5_r20m type=area use=val val=400
v.to.rast cdda_6_v out=cdda_6_r20m type=area use=val val=400
# Fill the nulls or the r.mapcalc command will not work: (attention this will create huge files!)
r.null map=cdda_1a_r20m null=0
r.null map=cdda_1b_r20m null=0
r.null map=cdda_2_r20m null=0
r.null map=cdda_3_r20m null=0
r.null map=cdda_4_r20m null=0
r.null map=cdda_5_r20m null=0
r.null map=cdda_6_r20m null=0

# Often there are overlapping, e.g. a national park that embeds a strict reserve
# Remove overlappings following the order between IUCN categories:
# 1a, 3, 4, 1b, 2, 5, 6
# Set cdda_6_r20m equal to zero if any of the other layer is different than zero
r.mapcalc 'cdda_6_r20m =                                \
  if(   cdda_1a_r20m == 0 && cdda_3_r20m  == 0   \
     && cdda_4_r20m  == 0 && cdda_1b_r20m == 0   \
     && cdda_2_r20m  == 0 && cdda_5_r20m  == 0,  \
     cdda_6_r20m,0)'
r.mapcalc 'cdda_5_r20m =                                \
  if(   cdda_1a_r20m == 0 && cdda_3_r20m  == 0   \
     && cdda_4_r20m  == 0 && cdda_1b_r20m == 0   \
     && cdda_2_r20m  == 0,                      \
     cdda_5_r20m,0)'
r.mapcalc 'cdda_2_r20m =                                \
  if(   cdda_1a_r20m == 0 && cdda_3_r20m  == 0   \
     && cdda_4_r20m  == 0 && cdda_1b_r20m == 0,  \
     cdda_2_r20m,0)'
r.mapcalc 'cdda_1b_r20m =                               \
  if(   cdda_1a_r20m == 0 && cdda_3_r20m  == 0   \
     && cdda_4_r20m  == 0,                      \
     cdda_1b_r20m,0)'
r.mapcalc 'cdda_4_r20m =                               \
  if(   cdda_1a_r20m == 0 && cdda_3_r20m  == 0,   \
     cdda_4_r20m,0)'
r.mapcalc 'cdda_3_r20m =    \
  if(   cdda_1a_r20m == 0,  \
     cdda_3_r20m,0)'

# Compute the area of each type of protected area within the 8x8km pixels
g.region -p res=8000
r.resamp.stats cdda_1a_r20m out=cdda_1a_r8km method=sum
r.resamp.stats cdda_1b_r20m out=cdda_1b_r8km method=sum
r.resamp.stats cdda_2_r20m out=cdda_2_r8km method=sum
r.resamp.stats cdda_3_r20m out=cdda_3_r8km method=sum
r.resamp.stats cdda_4_r20m out=cdda_4_r8km method=sum
r.resamp.stats cdda_5_r20m out=cdda_5_r8km method=sum
r.resamp.stats cdda_6_r20m out=cdda_6_r8km method=sum

# Compute the availability coefficient based on these coefficients:
# IUCN Protected Areas Categories System:
# Ia Strict Nature Reserve              0%
# Ib Wilderness Area                    10%
# II National Park                      80%
# III Natural Monument or Feature       10%
# IV Habitat/Species Management Area    10%
# V Protected Landscape/ Seascape       90%
# VI Protected area with sustainable use of natural resources 95%
# Altimetry:
# < 500             100%
# 500-1000          90%
# 1000-2000         70%
# > 2000            30%

r.mapcalc 'avalcoef = \
if(dtm_nonulls < 500, \
  1*( \
    0.0*cdda_1a_r8km+ \
    0.1*cdda_1b_r8km+ \
    0.8*cdda_2_r8km+ \
    0.1*cdda_3_r8km+ \
    0.1*cdda_4_r8km+ \
    0.9*cdda_5_r8km+ \
```

```
    0.95*cdda_6_r8km+ \
    1*(64000000-cdda_1a_r8km-cdda_1b_r8km-cdda_2_r8km-cdda_3_r8km-cdda_4_r8km-cdda_5_r8km-cdda_6_r8km) \
  )/64000000, \
  if(dtm_nonulls < 1000, \
    0.9*( \
    0.0*cdda_1a_r8km+ \
    0.1*cdda_1b_r8km+ \
    0.8*cdda_2_r8km+ \
    0.1*cdda_3_r8km+ \
    0.1*cdda_4_r8km+ \
    0.9*cdda_5_r8km+ \
    0.95*cdda_6_r8km+ \
    1*(64000000-cdda_1a_r8km-cdda_1b_r8km-cdda_2_r8km-cdda_3_r8km-cdda_4_r8km-cdda_5_r8km-cdda_6_r8km) \
  )/64000000, \
  if(dtm_nonulls < 2000, \
    0.7*( \
    0.0*cdda_1a_r8km+ \
    0.1*cdda_1b_r8km+ \
    0.8*cdda_2_r8km+ \
    0.1*cdda_3_r8km+ \
    0.1*cdda_4_r8km+ \
    0.9*cdda_5_r8km+ \
    0.95*cdda_6_r8km+ \
    1*(64000000-cdda_1a_r8km-cdda_1b_r8km-cdda_2_r8km-cdda_3_r8km-cdda_4_r8km-cdda_5_r8km-cdda_6_r8km) \
  )/64000000, \
    0.3*( \
    0.0*cdda_1a_r8km+ \
    0.1*cdda_1b_r8km+ \
    0.8*cdda_2_r8km+ \
    0.1*cdda_3_r8km+ \
    0.1*cdda_4_r8km+ \
    0.9*cdda_5_r8km+ \
    0.95*cdda_6_r8km+ \
    1*(64000000-cdda_1a_r8km-cdda_1b_r8km-cdda_2_r8km-cdda_3_r8km-cdda_4_r8km-cdda_5_r8km-cdda_6_r8km) \
  )/64000000 \
  )))'

# Export the map in a format readable by FFSM++ (a simple ASCII file)
r.out.ascii input=avalcoef output=/home/lobianco/git/ffsm_pp/data/gis/france/avalcoef.grd
```

Finally replace the NA values in the outputed file with the one used in FFSM++ (default: -9999).
