Batch functions for downloading and processing the 10-meter NED

Stephen Roecker

Wednesday, October 01, 2014

This document displays some R batch functions for downloading, mosaicing, warping, adding pyramids, and calculating terrain derivatives from the USGS seamless 10-meter NED (National Elevation Dataset), using the R package gdalUtils. As a bonus many of the raster outputs are tiled and compressed. This shrunk some rasters to a third of their original size, and also increased the rendering speed.

The batch commands are designed to run again the NED tile index, the NLCD dataset, and the SAPOLYGON layer for each MLRA office. Also it presumes a certain file organization structure, in order to write the files to their respective folders.

The primary workhorse of these batch functions is GDAL (Geospatial Data Abstraction Library). GDAL is a popular library for reading, writing, and converting various raster and vector formats, and is incorporated into most GIS software, including ArcGIS since version 10.0. The particuar version I used came included with QGIS (which is CCE approved).

Once you begin this sequence of commands will last several days. However it is not terribly memory intensize, so you should be able to work on other projects while it is running. The first thing that needs to be done, is loading the necessary R libraries. If they're not already installed you will have to do this the first time (e.g. "install.packages("gdalUtils", dependencies=TRUE)").

library(gdalUtils)  
library(foreign)  
library(rgdal)  
library(raster)

Next the proper GDAL path has to be set. The first location is the default path on my work computer, the second my personal computer. If this isn't set gdalUtils will do a brute force search of your computer, which usually finds GDAL 1.7 instead of the GDAL 10.1. The new version has additional features, which many these batch functions use.

gdal\_setInstallation(search\_path="C:/Program Files/QGIS Dufour/bin", rescan=T)  
gdal\_setInstallation(search\_path="C:/OSGeo4W64/bin", rescan=T)

Next numerous parameters need to be set which get used later by many of the functions or commands. Modify these file paths and lists as necessary. For example, I organized my files by "C:/geodata/project\_data/11ATL"", so 11 will have to replace by 10 or 2 for your respective Regions.

# Set parameters  
source("C:/Users/stephen.roecker/Documents/stools/trunk/gis/nedFunctions.R")  
source("C:/Users/stephen.roecker/Documents/stools/trunk/gis/gdalUtilsFunctions.R")  
tile.p <- "M:/geodata/elevation/ned/tiles/img/"  
office.l <- c("11ATL", "11AUR", "11CAR", "11CLI", "11FIN", "11GAL", "11IND", "11JUN", "11SPR", "11UNI", "11WAV")  
pd.p <- "M:/geodata/project\_data/"  
geodatabase <- paste(pd.p, office.l,"/11", office.l, ".gdb", sep="")  
nlcd.p <- "D:/geodata/land\_use\_land\_cover/nlcd\_2011\_landcover\_2011\_edition\_2014\_03\_31.img"  
nlcd.p.l <- paste(pd.p, office.l,"/nlcd", 30, "m\_11", office.l, "\_lulc2011.tif", sep="")  
crsarg <- CRSargs(CRS("+init=epsg:5070"))

## Download and unzip tile list

To start you need to download the 10-meter NED tiles from the USGS. Because the USGS manages these tiles using an ftp server it's easy for R download them one at a time with the following function. For whatever reason the downloading fuction doesn't work from RStudio, so this function needs to be run from the vanilla R console. Due to hickups in the downloading some files maynot download completely. I still need to write a function to identify corrupt or missing zip files.

ned.l<- makeNedList(geodatabase, office.l)  
download.l <- sort(unlist(ned.l))  
  
batchDownload(download.l)  
batchUnzip(ned.l)

## Subset NLCD by MLRA office

The NLCD layer is used as a standard coordinate reference system from which to warp the NED mosaics too, and for subseting by MLRA office.

# batchNlcdSubset(nlcd.p, nlcd.p.l)

## Mosaic tile list.

mo.l <- lapply(ned.l, function(x) paste(tile.p, "img", x, "\_13.img", sep=""))  
dst.p <- paste(pd.p, office.l[i], "/ned09d\_11", office.l[i], ".tif", sep="")  
  
mosaicList(mo.l, dst.p)

## Warp NED from a geographic to projected coordinate system

For warping from EPSG:4326 to EPSG:5070, I've used bilinear resampling which is my personal preference for some of the reasons discussed by Frank Warmerdam (<http://courses.neteler.org/gdal-raster-data-tips-and-tricks/>). For upscaling or aggregating the 10-meter to 30-meter DEM I use average resampling. Consequentially this makes the most sense and has been the approach used in several studies (Smith et al, 2006; Roecker and Thompson, 2010). Because DEM are later used for terrain analysis they are left uncompressed and untiled, which results in file sizes of approximately 10GB.

dem.l <- paste(pd.p, office.l, "/ned09d\_11", office.l, ".tif", sep="")  
  
batchWarp(dem.l, nlcd.p, "09d", "10m", "bilinear", crsarg)  
  
dem.l <- paste(pd.p, office.l, "/ned10m\_11", office.l, ".tif", sep="")  
  
batchWarpAverage(dem.l, "10m", "30m")  
  
dem30.tif <- paste(pd.p, office.l, "/ned30m\_11", office.l, ".tif", sep="")

## Calculate hillshade, slope, and aspect

GDALs DEM tools use Horn'n (1981) algorithms as the default, as does ArcInfo and GRASS.

dem.l <- paste(pd.p, office.l, "/ned10m\_11", office.l, ".tif", sep="")  
  
batchDEM(dem.l)

# Mosaic the 30-meter MLRA office mosaics into a Region office mosaic

nlcd.l <- list(nlcd.p)  
dst.p <- "C:/geodata/project\_data/REGION11/nlcd30m\_R11.tif"  
mosaicNlcdList(nlcd.l, dst.p)  
  
dem.l <- list(paste(pd.p, office.l, "/ned30m\_11", office.l, ".tif", sep=""))  
dem.p <- "C:/geodata/project\_data/Region11/ned30m\_R11.tif"  
mosaicList(dem.l, dem.p)  
batchTerrain(dem.p)  
  
hil.p <- paste(pd.p, office.l, "/", sep="")  
hil.p.l <- paste(hil.p, "ned10m\_11", office.l, "\_hillshade.tif", sep="")  
mosaicList(list(hil.p.l), "E:/geodata/project\_data/11REGION/ned10m\_11R\_hillshade.tif", "Byte", c("COMPRESS=DEFLATE", "TILED=YES", "BIGTIFF=YES"), 0)