RSAGA function instructions

Stephen Roecker

Wednesday, October 08, 2014

This document displays some R batch functions for generating DEM derivatives using the RSAGA R package. It is assumes the reader has already compiled a series of DEM following the nedInstrucitons document.

SAGA is an open-source GIS that was originally developed in 1996 as a terrain analysis toolbox, known as DiGem. Since then it has become a fully fledged GIS, with additional modules for vector geoprocessing, image analysis, and geostatistics. While not as well documented at GRASS or ArcGIS it offers an intuitive interface, and includes a range algorithms not found elsewhere. Through the use of the RSAGA package, SAGA modules can called from R, and workflows developed. Unlike other GIS, SAGA utilizes significant RAM instead of using file caching. This makes SAGA fast, but it can also overwhelm a computer if to many large rasters are loaded. However I've noticed when using a solid state drive (SSD) I can process rasters than exceded my 16GB of RAM for certain SAGA modules that only use small local neighorhoods.

To begin, the necessary libaries must be loaded, as well as the custom batch functions.

library(gdalUtils)  
library(RSAGA)  
  
source("C:/Users/Stephen/SkyDrive/R/gis/gdalUtilsFunctions.R")  
source("C:/Users/Stephen/SkyDrive/R/gis/rsagaFunctions.R")

Next the proper GDAL and RSAGA path has to be set. The first GDAL 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:/ProgramData/QGIS/QGISDufour/bin", rescan=T)  
gdal\_setInstallation(search\_path="C:/OSGeo4W64/bin", rescan=T)  
myenv <- rsaga.env(path="C:/Program Files (x86)/SAGA-GIS")

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.

setwd("E:/geodata/project\_data")  
pd.p <- "E:/geodata/project\_data/11"  
office.l <- c("ATL", "AUR", "CAR", "CLI", "FIN", "GAL", "IND", "JUN", "SPR", "UNI", "WAV")  
sdat.p <- paste(pd.p, office.l, "/sdat/", sep="")  
state <- c("IA", "IL", "IN", "KS", "KY", "MI", "MN", "MO", "NE", "OH", "OK", "SD", "WI")  
nhdgdb <- paste("E:/geodata/hydrography/NHDH\_", state, ".gdb", sep="")  
  
# Generater raster names  
dem10 <- paste(sdat.p, "ned10m\_11", office.l, ".sgrd", sep="")  
dem30 <- paste(sdat.p, "ned30m\_11", office.l, ".sgrd", sep="")  
radiusD <- 2  
radiusV <- round(1000/30/2-1, 0)  
  
res10 <- paste(strsplit(dem10, ".sgrd"), sep="")  
res30 <- paste(strsplit(dem30, ".sgrd"), sep="")  
g10 <- list(  
 slopeR=paste(res10, "\_slopeR", 1+2\*radiusD, sep=""),  
 slope=paste(res10, "\_slope", 1+2\*radiusD, sep=""),  
 slopeD=paste(res10, "\_slopeD", 1+2\*radiusD, sep=""),  
 aspect=paste(res10, "\_aspect", 1+2\*radiusD, sep=""),  
 cupro=paste(res10, "\_cupro", 1+2\*radiusD, sep=""),  
 cucon=paste(res10, "\_cucon", 1+2\*radiusD, sep=""),  
 cutan=paste(res10, "\_cutan", 1+2\*radiusD, sep=""),  
 cumax=paste(res10, "\_cumax", 1+2\*radiusD, sep=""),  
 cumin=paste(res10, "\_cumin", 1+2\*radiusD, sep="")  
 )  
g30 <- list(  
 elev=paste(res30, "\_elev", 1+2\*radiusD, sep=""),  
 slope=paste(res30, "\_slope", 1+2\*radiusD, sep=""),  
 slopeR=paste(res30, "\_slopeR", 1+2\*radiusD, sep=""),  
 slopeD=paste(res30, "\_sloped", 1+2\*radiusD, sep=""),  
 aspect=paste(res30, "\_aspect", 1+2\*radiusD, sep=""),  
 valleys=paste(res30, "\_mvalleys", sep=""),  
 summits=paste(res30, "\_summits", sep=""),  
 carea=paste(res30, "\_carea", sep=""),  
 cheight=paste(res30, "\_cheight", sep=""),  
 wetness=paste(res30, "\_wetness", sep=""),  
 strahler=paste(res30, "\_strahler", sep=""),  
 z2stream=paste(res30, "\_z2stream", sep="")  
 )

# Convert GTiff to SAGA  
dem10.tif <- paste(pd.p, office.l, "/ned10m\_11", office.l, ".tif", sep="")  
dem30.tif <- paste(pd.p, office.l, "/ned30m\_11", office.l, ".tif", sep="")  
dem10.sdat <- paste(strsplit(dem10, ".sgrd"), ".sdat", sep="")  
dem30.sdat <- paste(strsplit(dem30, ".sgrd"), ".sdat", sep="")  
  
gdal\_GTiff2SAGA(dem10.tif, dem10.sdat)  
gdal\_GTiff2SAGA(dem30.tif, dem30.sdat)  
gdal\_GTiff2SAGA("E:/geodata/project\_data/11REGION/ned30m\_R11.tif",  
 "E:/geodata/project\_data/11REGION/sdat/ned30m\_R11.sdat")

# Calculate local derivatives for 10-meter DEM  
attach(lapply(g10, function(x) paste(x, ".sgrd", sep="")))  
rsaga.d1(dem10, 2)  
rsaga.d2(dem10, 2)  
rsaga.d3(dem10, 2)  
# Converts to radians then percent, 57.29578=180/pi, degrees=radians\*180/pi  
rsaga.grid.calculus(slopeD, slopeD, slope, "tan(a\*(1/57.29578))\*100")  
# Rescales curvatures so they can be exported as UInt16 to save file size  
rsaga.grid.calculus(cupro, cupro, cupro, "10000\*a")  
rsaga.grid.calculus(cucon, cucon, cucon, "-10000\*a")  
rsaga.grid.calculus(cumin, cumin, cumin, "10000\*a")  
rsaga.grid.calculus(cumax, cumax, cumax, "10000\*a")  
rsaga.grid.calculus(cucon, slopeD, cutan, "a\*sin(b/57.29578)")  
  
  
# Calculate regional derivatives for 30-meter DEM  
attach(lapply(g30, function(x) paste(x, ".sgrd", sep="")))  
rsaga.d0(dem30, 2)  
rsaga.d1(dem30, 2)  
rsaga.grid.calculus(slopeD, slopeD, slope, "tan(a\*(1/57.29578))\*100")  
rsaga.grid.calculus(slopeD, slopeD, slopeR, "a\*(1/57.29578)")  
rsaga.mrvbf(dem30, valleys, summits)  
  
# apply mask manually before  
elev.sdat <- paste(strsplit(elev, ".sgrd"), ".sdat", sep="")  
mosaicList(list(elev.sdat), "E:/geodata/project\_data/REGION11/ned30m\_R11\_elev5.tif")  
  
gdal\_translate(  
 src\_dataset="E:/geodata/project\_data/REGION11/ned30m\_R11\_elev5.tif",  
 dst\_dataset="E:/geodata/project\_data/REGION11/ned30m\_R11\_elev5\_masked.tif",  
 overwrite=TRUE,  
 verbose=TRUE  
)  
  
  
nhdshp <- paste(strsplit(nhdgdb, ".gdb"), "\_wb.shp", sep="")  
  
for(i in seq(nhdgdb)){  
 ogr2ogr(  
 src\_datasource\_name=nhdgdb[i],  
 dst\_datasource\_name=nhdshp[i],  
 layer="NHDWaterbody",  
 t\_srs="EPSG:5070",  
 overwrite=TRUE,  
 verbose=TRUE,  
 progress=TRUE)  
}  
  
# Seems to take exceptionally long for the States touching the Great Lakes, particularly MI. Best to run these States separately from OSGeo4W Shell to monitor their progress or do manually in SAGA.  
for(i in seq(nhdshp)){  
 cat(paste(format(Sys.time(), "%Y-%m-%d %H:%M:%S"), "burning", nhdshp[i], "\n"))  
 gdal\_rasterize(  
 src\_datasource=paste("E:/geodata/hydrography", sep=""),  
 dst\_filename="E:/geodata/project\_data/REGION11/ned30m\_R11\_elev5\_masked.tif",  
 l=paste("NHDH\_", state[i], "\_wb", sep=""),  
 where="AreaSqKm > 0.04047",  
 b=1,  
 burn=-99999,  
 verbose=TRUE  
 )   
}  
  
  
dem.sdat<- paste(strsplit(dem30, ".sgrd"), "\_elev5\_masked.sgrd", sep="")  
batchSubsetSAGA("E:/geodata/project\_data/REGION11/ned30m\_R11\_elev5\_masked.tif", dem.sdat, nlcdpath)  
  
dem.l<- paste(strsplit(dem30, ".sgrd"), "\_elev5\_masked.sgrd", sep="")  
rsaga.fill(dem.l)  
dem.l<- paste(strsplit(dem30, ".sgrd"), "\_elev5\_masked\_filled.sgrd", sep="")  
rsaga.ca(dem.l)  
rsaga.twi(slopeR, carea)  
rsaga.strahler(dem.l,4)  
rsaga.reclassify(strahler, -3, -1)  
  
strahler.rc<- paste(strsplit(dem30, ".sgrd"), "\_strahler\_rc0.sgrd", sep="")  
rsaga.ofd(dem.l, 1)  
  
# Write SAGA to GTiff  
attach(lapply(g10, function(x) paste(x, ".sdat", sep="")))  
int16.sdat <- c(slope)  
int16.tif <- unlist(lapply(strsplit(int16.sdat, "/sdat"), paste, collapse="", sep=""))  
int16.tif <- paste(strsplit(int16.tif, ".sdat"), ".tif", sep="")  
gdal\_SAGA2GTiff(int16.sdat, int16.tif, "Int16", -32768)  
  
slopeshape <- paste(pd.p, office.l, "/ned10m\_11", office.l, "\_slopeshape.tif", sep="")  
test <- list(cupro, cutan, slope)  
gdal.stack(test, slopeshape, "Int16", -32768)  
  
attach(lapply(g30, function(x) paste(x, ".sdat", sep="")))  
flt.sdat <- c(z2stream, valleys, wetness)  
flt.tif <- unlist(lapply(strsplit(flt.sdat, "/sdat"), paste, collapse="", sep=""))  
flt.tif <- paste(strsplit(flt.tif, ".sdat"), ".tif", sep="")  
gdal\_SAGA2GTiff(flt.sdat, flt.tif, "Float32", -99999)  
  
  
mosaicList(list(slope), "E:/geodata/project\_data/11REGION/ned10m\_11R\_slope5.tif", "Int16", c("COMPRESS=DEFLATE", "TILED=YES", "BIGTIFF=YES"), -32768)