Lionshead WildFire and compariosn between dNBR vs RdNBR

Mir Mazedur Rahaman

2/7/2021

# Loading required package

# Loading the MetaData and Raster file

setwd("C:/FIT/3rd Semester/RS-3/Code/WildFire\_RdNBR/")  
AOI <- readOGR("data/AOI.shp")  
# Pre fire Raster and metaData  
metaData\_pre <- readMeta("data/LC08\_L1TP\_045029\_20200901\_20200906\_01\_T1/LC08\_L1TP\_045029\_20200901\_20200906\_01\_T1\_MTL.txt")  
metaStack\_pre <- stackMeta(metaData\_pre)  
plot(metaStack\_pre)  
# Post fire Raster and metaData  
metaData\_post <- readMeta("data/LC08\_L1TP\_045029\_20201003\_20201015\_01\_T1/LC08\_L1TP\_045029\_20201003\_20201015\_01\_T1\_MTL.txt")  
metaStack\_post <- stackMeta(metaData\_post)  
plot(metaStack\_post)

# Image Pre Processing

# DOS Correction pre fire images and saving stack raster as geoTiff  
LS8\_pre\_dos <- radCor(metaStack\_pre, metaData\_pre, method = "dos", bandSet = c(2,3,4,5,7))  
writeRaster(LS8\_pre\_dos, filename="output/LS8\_pre\_dos.tif", bandorder='BIL', overwrite=TRUE)  
# DOS Correction post fire images and saving stack raster as geoTiff  
LS8\_post\_dos <- radCor(metaStack\_post, metaData\_post, method = "dos", bandSet = c(2,3,4,5,7))  
writeRaster(LS8\_post\_dos, filename="output/LS8\_post\_dos.tif", bandorder='BIL', overwrite=TRUE)  
# Load the processed raster  
LS8\_pre <- brick("output/LS8\_pre\_dos.tif")  
LS8\_post <- brick("output/LS8\_post\_dos.tif")  
# Plotting the RGB images of Pre fire and Post fire area  
plotRGB(LS8\_pre\_dos, r=3, g=2, b=1, stretch="lin")  
plotRGB(LS8\_post\_dos, r=3, g=2, b=1, stretch="lin")

# Calculating pre and post NBR and dNBR

# Function for NBR calculation  
nbr <- function(img){  
 br <- (img[[4]]-img[[5]])/(img[[4]]+img[[5]])  
 return(br)  
}  
# set the new projection and crop by the extent  
crs <- projection(LS8\_pre)  
crs  
AOI\_reproj <- spTransform(AOI,crs)  
# Calculating pre fire NBR  
nbr\_LS8\_pre <- calc(LS8\_pre, fun = nbr)  
# Crop by the extent  
nbr\_LS8\_pre\_crop <- crop(nbr\_LS8\_pre,extent(AOI\_reproj))  
nbr\_LS8\_pre\_crop <- mask(nbr\_LS8\_pre\_crop, AOI\_reproj)  
# Calculating post fire NBR  
nbr\_LS8\_post <- calc(LS8\_post, fun = nbr)  
nbr\_LS8\_post\_crop <- crop(nbr\_LS8\_post,extent(AOI\_reproj))  
nbr\_LS8\_post\_crop <- mask(nbr\_LS8\_post\_crop, AOI\_reproj)  
# dNBR  
dnbr <- (nbr\_LS8\_pre\_crop - nbr\_LS8\_post\_crop)  
# Plot the dNBR raster and save as jpeg  
jpeg('output/dNBR.jpg',width = 1000,   
 height = 700, res=200, units = "px", quality = 100, pointsize=10)  
par(mar=c(4,8,4,4))  
plot(dnbr, col = rev(terrain.colors(10)), main = 'dNBR')  
dev.off()  
# Saving the dNBR as tif  
writeRaster(dnbr, filename = "output/dNBR.tif", format="GTiff", overwrite=T)

# Calculating RdNBR

# RdNBR  
rdnbr <- (nbr\_LS8\_pre\_crop - nbr\_LS8\_post\_crop)/sqrt(abs(nbr\_LS8\_pre\_crop))  
# Rescaling the min and max  
min <- -2  
max <- 2  
rdnbr[rdnbr <= min] <- NA  
rdnbr[rdnbr >= max] <- NA  
# Plot the RdNBR raster and save as jpeg  
jpeg('output/RdNBR.jpg',width = 1000,   
 height = 700, res=200, units = "px", quality = 100, pointsize=10)  
par(mar=c(4,8,4,4))  
plot(rdnbr, col = rev(terrain.colors(10)), main = 'RdNBR')  
dev.off()  
# Saving the RdNBR as tif  
writeRaster(rdnbr, filename = "output/RdNBR.tif", format="GTiff", overwrite=T)

# Classification of dNBR according to CBI threshold

# Sets the ranges that will be used to classify dNBR information about the ranges used  
reclass\_df <- c(-Inf, -0.5, 0, # NA values  
 -0.5, -0.251, 1, # Enhanced Regrowth, High  
 -0.251, -0.101, 2, # Enhanced Regrowth, Low  
 -0.101, 0.099, 3, # Unburned  
 0.099, 0.269, 4, # Low-severity  
 0.269, 0.439, 5, # Moderate-low Severity  
 0.439, 0.659, 6, # Moderate-High Severity  
 0.659, 1.300, 7, # High-severity  
 1.300, +Inf, 0) # NA values  
# Sets a classification matrix  
reclass\_m <- matrix(reclass\_df, ncol=3, byrow=TRUE)  
# Classification matrix is used to classify dNBR\_scaled  
dnbr\_reclass <- reclassify(dnbr, reclass\_m, right = NA)  
# Build the legend for the burn severity map  
legend\_val <- c("NA values",  
 "Enhanced Regrowth, High",  
 "Enhanced Regrowth, Low",  
 "Unburned",  
 "Low-severity",  
 "Moderate-low Severity",  
 "Moderate-High Severity",  
 "High-severity"  
 )  
# Setting the colors for the severity map  
col\_val <- viridis(8, option = "A")  
# Plots the burn severity map  
jpeg('output/dNBR\_Reclass.jpg',width = 1300,   
 height = 700, res=200, units = "px", quality = 100, pointsize=10)  
par(mar=c(4,8,4,4))  
plot(dnbr\_reclass,  
 col = col\_val,  
 axes = FALSE,  
 box = FALSE,  
 legend = FALSE,  
 main = "Burn Severity Map of Lionshead from dNBR")  
legend('topright',  
 legend = legend\_val ,fill=col\_val,  
 inset=c(-0.25,0), pt.cex=0.7,cex=0.7,  
 xpd = TRUE, bty='n')  
dev.off()  
# Saving the dNBR as tif  
writeRaster(dnbr\_reclass, filename = "output/dNBR\_Reclassified.tif", format="GTiff", overwrite=TRUE)

# Classification of RdNBR according to CBI threshold

# Sets the ranges that will be used to classify dNBR information about the ranges used  
reclass\_df <- c(-Inf, -0.5, 0, # NA values  
 -0.5, -0.251, 1, # Enhanced Regrowth, High  
 -0.251, -0.101, 2, # Enhanced Regrowth, Low  
 -0.101, 0.099, 3, # Unburned  
 0.099, 0.269, 4, # Low-severity  
 0.269, 0.439, 5, # Moderate-low Severity  
 0.439, 0.659, 6, # Moderate-High Severity  
 0.659, 1.300, 7, # High-severity  
 1.300, +Inf, 0) # NA values  
# Sets a classification matrix  
reclass\_m <- matrix(reclass\_df, ncol=3, byrow=TRUE)  
# Classification matrix is used to classify RdNBR\_scaled  
rdnbr\_reclass <- reclassify(rdnbr, reclass\_m, right = NA)  
# Build the legend for the burn severity map  
legend\_val <- c("NA values",  
 "Enhanced Regrowth, High",  
 "Enhanced Regrowth, Low",  
 "Unburned",  
 "Low-severity",  
 "Moderate-low Severity",  
 "Moderate-High Severity",  
 "High-severity"  
 )  
# Setting the colors for the severity map  
col\_val <- viridis(8, option = "A")  
# Plots the burn severity map  
jpeg('output/RdNBR\_Reclass.jpg',width = 1300,   
 height = 700, res=200, units = "px", quality = 100, pointsize=10)  
par(mar=c(4,8,4,4))  
plot(rdnbr\_reclass,  
 col = col\_val,  
 axes = FALSE,  
 box = FALSE,  
 legend = FALSE,  
 main = "Burn Severity Map of Lionshead from RdNBR")  
legend('topright',  
 legend = legend\_val ,fill=col\_val,  
 inset=c(-0.25,0), pt.cex=0.7,cex=0.7,  
 xpd = TRUE, bty='n')  
dev.off()  
# Saving the dNBR as tif  
writeRaster(rdnbr\_reclass, filename = "output/RdNBR\_Reclassified.tif", format="GTiff", overwrite=TRUE)

# Result analysis, Graph and Chart

# Area of each Burn Severity class for dNBR  
Area\_dNBR <- raster::extract(dnbr\_reclass, AOI\_reproj,df = TRUE)  
Area\_dNBR %>%  
 group\_by(layer) %>%  
 summarize(count = n(), area\_meters = n() \* (30 \* 30))  
  
# Pie chart of dNBR class and its Proportions  
pie(table(Area\_dNBR), labels = paste(round(prop.table(table(Area\_dNBR))\*100), "%", sep = ""), col = col\_val, main = "Proportions of dNBR Classes")  
legend('topright',  
 legend = legend\_val ,fill=col\_val,  
 inset=c(-0.05,0),  
 pt.cex=0.7,cex=0.7,  
 xpd = TRUE, bty='n')  
  
# Area of each Burn Severity class for RdNBR  
Area\_dNBR <- raster::extract(rdnbr\_reclass, AOI\_reproj,df = TRUE)  
Area\_dNBR %>%  
 group\_by(layer) %>%  
 summarize(count = n(), area\_meters = n() \* (30 \* 30))  
  
# Pie chart of RdNBR class and its Proportions  
pie(table(RdNBR), labels = paste(round(prop.table(table(RdNBR))\*100), "%", sep = ""), col = col\_val, main = "Proportions of dNBR Classes")  
legend('topright',  
 legend = legend\_val ,fill=col\_val,  
 inset=c(-0.05,0),  
 pt.cex=0.7,cex=0.7,  
 xpd = TRUE, bty='n')