

Example code to calculate landscape metrics for high severity pattern

Caralyn Gorman

```
library(landscapemetrics)
library(raster)
library(tidyverse)
library(dplyr)
library(plyr)
library(sf)
library(rgdal)
library(tmap)
library(ggplot2)
library(ggpubr)
library(data.table)
library(knitr)
```

These functions are used to calculate landscape metrics examined in this study. Other landscape metrics within the `landscapemetrics` package can be substituted in the `lms_function`.

```
# function for calculating multiple metrics from the `landscapemetrics` package
lsm_function <- function(raster1) {
  calculate_lsm(raster1,
    what = c(
      "lsm_c_tca",
      "lsm_c_clumpy",
      "lsm_c_cohesion",
      "lsm_c_pd",
      "lsm_c_pladj",
      "lsm_c_pland"),
    full_name = TRUE)
}

# function for calculating area-weighted mean patch area
weighted_mean <- function(raster_pic) {
  patch_table <- lsm_p_area(raster_pic, directions = 4) %>% filter(class == 4)
  area_acres <- (patch_table$value) * 2.47105
  total_area <- sum(area_acres)
  weighted_patch <- ((area_acres)^2)/total_area
  sum(weighted_patch)
}

# function for calculating area-weighted mean core area
weighted_core_mean <- function(raster_pic) {
  patch_table <- lsm_p_core(raster_pic, directions = 4, edge_depth = 5) %>%
    filter(class == 4)
  area_acres <- (patch_table$value) * 2.47105
  total_area <- sum(area_acres)
  weighted_patch <- ((area_acres)^2)/total_area
  sum(weighted_patch)
}
```

```
# function for calculating perimeter-area ratio
mean_paras <- function(raster_pic) {
  mean_paras <- lsm_c_para_mn(raster_pic, directions = 4) %>%
    filter(class == 4)
  mean_paras_value <- mean_paras$value
  mean_paras_value
}
```

This methodology requires a .csv file including identifying information about each fire (such as MTBS Fire_Name, Fire_ID, Acres, etc), as well as .tif files of categorical burn severity (typically in MTBS data bundles as dnbr6.tif) downloaded from MTBS. An example dataset, which features five wildfires in California, is included in this repository.

First, load your .csv:

```
# for public users:
# fires_list_csv <- read.csv("../example_code_data/example_csv.csv")

# on Caralyn's personal computer:
fires_list_csv <- read.csv("~/Documents/example_code_data/example_csv.csv")
```

Then, specify the path on your local machine to a folder containing only the .tif files. This code identifies each file and reads each as a raster:

```
# for public users:
# dir <- "../example_code_data/dnbr6tifs/"

# on Caralyn's personal computer:
dir <- "~/Documents/example_code_data/dnbr6tifs/"
files <- list.files(path = dir, pattern = ".tif")
path <- paste0(dir,files)
rasters <- lapply(paste0(dir, files), raster)
```

This code will apply the lsm_function to each raster file, calculating most landscape metrics chosen in this study. It then binds the information from all dataframes created to create a single dataframe with information for each fire. You can optionally write this raw data to a .csv.

```
many_tables <- lapply(rasters, lsm_function)
many_tables_ID <- Map(cbind, many_tables, ID = files)
new_tbl <- bind_rows(many_tables_ID, .id = "column_label")
# write.csv(new_tbl_ss, "~/your/path/here/")
# new_tbl_ss <- read.csv("~/your/path/here/")
```

This code reshapes the data to put it in a more usable format:

```
value_name_id <- new_tbl %>% filter(class == 4) %>% select(value, name, ID)
reshaped <- reshape(value_name_id, timevar = "name", idvar = "ID", direction = "wide")
reshaped$new_ID <- stringr::str_replace(reshaped$ID, '\\_dnbr6.tif', '')
reshaped$new_ID = substr(reshaped$new_ID, 1, nchar(reshaped$new_ID)-18)
reshaped$new_ID = toupper(reshaped$new_ID)
names(reshaped)[names(reshaped) == "new_ID"] <- "Fire_ID"
```

This chunk calculates the area-weighted mean patch area and creates a dataframe of results:

```
patch_tables <- lapply(rasters, weighted_mean)
weighted_mean_patchtbl <- as.data.frame(unlist(patch_tables))
weighted_df <- data.frame(weighted_patch_area = unlist(patch_tables), Fire_ID = fires_list_csv$Fire_ID)
```

This code calculates area-weight mean core area and creates a dataframe of results:

```
core_tables <- lapply(rasters, weighted_core_mean)
weighted_mean_coretbl <- as.data.frame(unlist(core_tables))
weighted_core_df <- data.frame(weighted_core_area = unlist(core_tables), Fire_ID = fires_list_csv$Fire_ID)
```

This code calculates mean perimeter-area ratio and creates a dataframe of results:

```
para_tables <- lapply(rasters, mean_paras)
weighted_para_ttbl <- as.data.frame(unlist(para_tables))
weighted_para_df <- data.frame(mean_para = unlist(para_tables), Fire_ID = fires_list_csv$Fire_ID)
```

Join the landscape metrics data to your original list of fires, and reset the column names:

```
big_ttbl <- join_all(list(reshaped, fires_list_csv, weighted_df, weighted_core_df, weighted_para_df), by = "Fire_ID")
column_names <- colnames(big_ttbl)
setnames(big_ttbl, old = column_names, new = c("total_class_area", "clumpiness", "pci", "patch_density", "pladj", "pland", "Fire_ID", "Fire_Name"))
```

Final dataframe:

Table 1: Dataframe

total_class_area	clumpiness	pci	patch_density	pladj	pland	Fire_ID	Fire_Name
0.8890185	97.15110	0.3096117	87.16151	1.9870668	126.90	CA3552511846520120804	PIUTE CO
0.8293906	96.13320	0.5023106	82.08383	1.8527971	258.66	CA3555411854220100912	CANYON
0.9134770	99.30710	0.5732025	91.40946	9.1832831	1107.81	CA3562311808819970809	JACKS
0.6121625	86.32410	0.1998002	57.17391	0.2297702	2.25	CA3578311796620100716	SAND 2
0.5935872	84.32893	1.3215463	57.76042	0.8097986	17.64	CA3651711910820050707	MILLWOC

This workflow can be repeated on multiple groups of fires, and comparisons can be made using Wilcoxon Rank Sum tests via `pairwise.wilcox.test()`.