Remote sensing in rangeland fire ecology: Comparing imagery to measured fire behavior, and burn severity across prescribed burns and wildfires

Supplementary Information: script for R and Copernicus browser

DA McGranahan

Contents

This document is organized around the principal steps used to wrangle and analyze data in this project, and summarizes the R script used to perform each step. Script is provided for readers to understand the means by which data were sourced, assembled, and analyzed to better contextualize the results and their interpretations in the main manuscript. While in some cases it might serve as a resource for solutions to the data wrangling and analysis problems encountered in this project, is not presented as a linear, start-to-finish set of reproducible code. Some steps rely on data manipulated externally in QGIS.

Setup

```
# Load necessary packages via pacman utility
  pacman::p_load(tidyverse, magrittr, readxl, # data wrangling
                 foreach, doSNOW,             # parallel processing
                 sf, terra,                   # Spatial data
                 tigris,                      # Access US census data
                 climateR,                    # Fetch TerraClim data
                 lme4, emmeans       )        # statistical analysis
```

Study region

```
# bounding box for focal area of study
  region_box <-
```

```
    tibble(feature = 'region',
           Easting = c(-900000, -215000, -215000, -900000),
           Northing = c(2470000, 2950000, 2470000, 2950000) ) %>%
    st_as_sf(coords = c("Easting", "Northing"), crs = 102039) %>%
    group_by(feature) %>%
    summarize(geometry = st_combine(geometry)) %>%
    st_cast("POLYGON") %>%
    st_bbox() %>%
    st_as_sfc(st_bbox(.))
# EPA Level III ecoregions
# (https://www.epa.gov/eco-research/level-iii-and-iv-ecoregions-continental-united-states)
  ngp <-
    read_sf(local_dir, # downloaded to local directory
            "us_eco_l3_state_boundaries") %>%
      filter(NA_L3NAME %in% c('Northwestern Glaciated Plains',
                              'Northwestern Great Plains')) %>%
      select(NA_L3NAME, STATE_NAME) %>%
      rename(L3 = NA_L3NAME, state = STATE_NAME) %>%
     st_crop(region_box)
# Get counties for the region
  ngp_counties <-
    tigris::counties(state = c('MT', "ND", 'SD'), cb = T) %>%
    st_transform(st_crs(ngp)) %>%
    st_intersection(ngp)
```

Data aquisition

Regional precipitation and anomalies

```
# Create grid for region
  ngp_grid <-
    ngp %>%
      st_make_grid(cellsize = c(10000, 10000)) %>%
      st_intersection(ngp)  %>%
      st_as_sf() %>%
      rowid_to_column('cell')
# PReciptation data for study period
{
  # set up for parallel processing
  cores= parallel::detectCores()
  cl <- makeCluster(cores)
  registerDoSNOW(cl )

  begin = Sys.time()
  ngp_ppt <-
    foreach(i=1:length(ngp_grid$cell),
            .combine = bind_rows,
            .errorhandling = 'remove',
            .inorder = FALSE,
            .packages = c('tidyverse', 'sf', 'climateR')) %dopar% {
              getTerraClim(
                AOI = ngp_grid %>%
                      slice(i) %>%
```

```
                    st_centroid() ,
                varname = 'ppt',
                startDate = '2017-01-01',
                endDate = '2024-12-31'  ) %>%
                as_tibble() %>%
                separate(date, into = c('year', 'month', 'day'), sep = "-") %>%
                 group_by(year) %>%
                summarize(ppt = sum(ppt_total, na.rm = TRUE)) %>%
                summarize(ppt = mean(ppt, na.rm = TRUE)) %>%
                add_column(cell = i)
            }
  stopCluster(cl)
  Sys.time() - begin
}

  ppt_dat <- lst()
  ppt_grd <-
    ngp_grid  %>%
      merge(by = 'cell',
                ngp_ppt)
  ppt_dat$ppt_grd <- ppt_grd
# Find precip for counties with Research Extension Centers
  ppt_dat$REC_PPT <-
  ppt_grd %>%
    st_intersection(ngp_counties %>%
                      filter(STUSPS == 'ND',
                             NAME %in% c('Stutsman', 'Kidder', 'Adams')) %>%
                      select(L3) ) %>%
  filter(st_geometry_type(., by_geometry = TRUE) %in% c('MULTIPOLYGON', 'POLYGON'))  %>%
    as_tibble() %>%
    group_by(L3) %>%
    summarize(PPT_Mean = mean(ppt),
              PPT_SD = sd(ppt))
# Assign anomalies to grid
  SDs = 3 # define anomaly by standard deviation
  ppt_dat$anomaly <-
    ppt_grd %>%
      st_intersection(ngp, .) %>%
      filter(st_geometry_type(., by_geometry=TRUE) %in% c('MULTIPOLYGON','POLYGON')) %>%
      mutate(anomaly = case_when(
        L3 == "Northwestern Glaciated Plains" ~ ppt - 447,
        L3 == "Northwestern Great Plains" ~ ppt - 431 ),
        anom_cat = case_when(
          L3 == "Northwestern Glaciated Plains" &
            abs(anomaly) > 12 * SDs ~ 'beyond',
          L3 == "Northwestern Glaciated Plains" &
            abs(anomaly) < 12 * SDs ~ 'within',
          L3 == "Northwestern Great Plains" &
            abs(anomaly) > 19 * SDs ~ 'beyond',
          L3 == "Northwestern Great Plains" &
            abs(anomaly) < 19 * SDs ~ 'within') )
```

Wildfire perimeters

```
# Get fire perimeters for the study region by L3
  # https://data-nifc.opendata.arcgis.com/datasets/
  # nifc::interagencyfireperimeterhistory-all-years-view/
region_perims2 <-
  read_sf(local_dir, # downloaded to local directory
          "InterAgencyFirePerimeterHistory_All_Years_View") %>%
    select(FIRE_YEAR, INCIDENT, FEATURE_CA) %>%
    st_transform(st_crs(l3)) %>%
    st_make_valid(.) %>%
    st_intersection(l3) %>%
    rename(FireYear = FIRE_YEAR, FireName = INCIDENT, FireType = FEATURE_CA)

region_perims <-
  region_perims2  %>%
  mutate(FireType = case_when(
          str_sub(FireType, 1,4)=='Wild' ~ 'Wildfire',
          TRUE ~ 'RxFire' )) %>%
  filter(FireType == 'Wildfire',
         between(as.numeric(FireYear), 2017, 2024),
         state != "Wyoming") %>%
  mutate(FireName = case_when(
              FireName == 'nd-crr-fy22-wf-knutsen' ~ 'Knutsen',
              TRUE ~ FireName )) %>%
  mutate(FireCode = str_remove_all(FireName, ' Fire'),
         FireCode = gsub(pattern     = "[^a-z,A-Z,0-9]",
                         replacement = "",
                         FireCode),
         FireCode = paste0(FireCode, '_', FireYear),
         Ha = st_area(.),
         Ha = as.numeric(Ha) * 0.0001) %>%
  group_by(FireCode, FireYear, FireName, FireType, L3, state) %>%
  summarize(AreaHa = sum(Ha),
            .groups ='drop')  %>%
  group_by(FireCode) %>%
  arrange(desc(AreaHa)) %>%
  slice(1) %>% # in case fire crosses ecoregion/state lines
  ungroup()
# Identify wildfires within the precipitation anomaly
  win_fires <-
    region_perims %>%
      st_centroid() %>%
      st_intersection(ppt_dat$anomaly) %>%
      filter(anom_cat == 'within',
             AreaHa > 10) %>%
    as_tibble() %>%
    select(FireCode, L3, state, anomaly)

  precip_perims <-
    region_perims %>%
    filter(FireCode %in% win_fires$FireCode)
# Get rangeland classifications
  # Locally-saved USFS Extent of US Rangelands raster product
```

```
    # https://apps.fs.usda.gov/arcx/rest/services/
    # RDW_LandscapeAndWildlife/Extent_of_US_Rangelands/MapServer
  rr_tr <- terra::rast(paste0(local_dir, "/USrangelands.tif")) %>%
            terra::crop(st_transform(region_perims, 5070))
# Find the proportion rangeland for each fire
  range_fires <- tibble()
  for(i in 1:length(unique(precip_perims$FireCode))){
    fire = unique(precip_perims$FireCode)[i]
    precip_perims %>%
      filter(FireCode == fire )%>%
      st_transform(st_crs(rr_tr)) %>%
      terra::extract(rr_tr, .) %>%
      group_by(LABEL) %>%
      summarize(pixels = n() ) %>%
      mutate(prop = pixels / sum(pixels)) %>%
      add_column(FireCode = fire, .before = 1) %>%
      filter(LABEL == 'Rangeland') %>%
      bind_rows(range_fires) -> range_fires }
  range_fires %<>%
    mutate(RangeHa = (pixels * 900)* 0.0001) %>%
    select(-LABEL, -pixels)  %>%
    rename(PropRange = prop)
# create comparison wildfires object
  gp_perims <-
    bind_rows(
    # NW Glaciated Plains
      precip_perims %>%
        right_join(by = 'FireCode',
                   range_fires) %>%
        filter(L3 == 'Northwestern Glaciated Plains') %>%
        arrange(desc(PropRange)) %>%
        filter(PropRange > 0.2),
    # NW Great Plains
      precip_perims %>%
        right_join(by = 'FireCode',
                   range_fires) %>%
        merge(by ='FireCode',
              win_fires %>% select(FireCode, anomaly)) %>%
        filter(L3 == 'Northwestern Great Plains' &
               RangeHa > 10) %>%
        arrange((abs(anomaly)), desc(PropRange)) %>%
        slice(1:20) ) %>%
     mutate(type = 'Wildfire',
                zone = case_when(
                  FireCode %in% c('Swather_2021','Hwy31101STWest_2022', 'CB00121_2021',
                                  'CoalSeamWest_2021', '1806HunkpapaCreek_2021') ~
                    'east',
                  L3 == 'Northwestern Great Plains' ~ 'west',
                  L3 == 'Northwestern Glaciated Plains' ~ 'east'))
```

Remotely-sensed data

Retrieving imagery

This script was used to create and export rectangular Area of Interest (AOI) `.kml` files for each wildfire for input into Copernicus browser.

```
  for(i in 1:length(unique(gp_perims$FireCode))) {
    wd = "./AOIs"
    fire = unique(gp_perims$FireCode)[i]
    gp_perims %>%
      filter(FireCode == fire) %>%
      st_transform(4326) %>%
      st_bbox() %>%
      st_as_sfc() %>%
      st_write(., paste0(wd, '/', fire, '.kml'), append = FALSE )}
```

This EvalScript for the Copernicus browser exports Normalized Burn Ratio (NBR) and Normalized Differenced Vegetation Index (NDVI) for the active AOI.

```
//VERSION=3
function setup() {
  return {
    input: ["B04", "B08", "B12"],
    output: { bands: 2, sampleType: "UINT16" }
  };
}

function evaluatePixel(sample) {
  let nbr = index(sample.B08, sample.B12);
  let ndvi = index(sample.B08, sample.B04);
  // apply offset for UINT16
  return [10000 * nbr + 10000,
          10000 * ndvi + 10000];
}
```

Processing imagery

```
  # load perimeters after editing in QGIS
    wf_sf <- read_sf('./SpatialData.gpkg', 'GreatPlainsModifiedWF')

    wf_dir = 'Folder/With/Sentinel/Imagery'

    wf_images <-
      tibble(file = list.files(wf_dir, pattern = "\\.tiff$", ignore.case = TRUE) ) %>%
      mutate(info = str_remove(file, '.tiff')) %>%
      separate(info, into = c('FireCode', 'period', 'ImageDate'), sep = '_') %>%
      mutate(FireCode = str_replace(FireCode, '-', '_'))

    wildfires <- wf_sf %>%
                filter(FireCode %in% unique(wf_images$FireCode) ) %>%
                st_transform(albersEAC) %>%
                mutate(AreaHa = st_area(.),
                       AreaHa = as.numeric(AreaHa) * 0.0001)
```

```r
# Sample raster imagery
 WfBurnIndices <- tibble()
  for(i in 1:length(wildfires$FireCode)) {
    # Get fire
      fire <-  wildfires %>%
                filter(FireCode == unique(wildfires$FireCode)[i]) %>%
                st_transform(st_crs(r_rr))
    # create gridded sample points
      buff <- fire  %>% st_buffer(-20)
      # cell size factor scaled to total area
        cs = case_when(
          fire$AreaHa < 10 ~ 25,
          between(fire$AreaHa, 10, 50) ~ 36,
          between(fire$AreaHa, 50, 100) ~ 50,
          between(fire$AreaHa, 100, 200) ~ 100,
          between(fire$AreaHa, 200, 300) ~ 125,
          between(fire$AreaHa, 300, 500) ~ 150,
          between(fire$AreaHa, 500, 700) ~ 200,
          between(fire$AreaHa, 700, 3000) ~ 300,
          between(fire$AreaHa, 3000, 5000) ~ 500,
          between(fire$AreaHa, 5000, 7000) ~ 700,
          fire$AreaHa > 7000 ~ 1000 )
      pts <- buff %>%
        st_make_grid(cellsize = cs,
                     square = FALSE) %>%
        st_centroid() %>%
        st_intersection(buff) %>%
        st_as_sf() %>%
        rowid_to_column('ID')
     # Filter sample points to rangeland cells
        r_pts <-
          terra::extract(rr_tr,
                         terra::vect(pts),
                         df = TRUE) %>%
          filter(LABEL %in% c('Rangeland',
                              'Transitional Rangeland',
                              'Afforested CO') )
        pts %<>% filter(ID %in% r_pts$ID) %>%
                  st_transform(4326)
  # dNBR & NDVI
    # Fetch & process multi-band rasters
    # pre image
      pre_image = filter(wf_images, FireCode == fire$FireCode, ,
                         period == 'B')$file
      pre_path = paste0(wf_dir, '/', pre_image)
      pre_ras <- terra::rast(pre_path) %>%
                 terra::crop(terra::vect(fire %>% st_transform(4326)))
      pre_ras <-  (pre_ras-10000)/10000
    # post-fire
      post_image = filter(wf_images, FireCode == fire$FireCode, ,
                          period == 'A')$file
      post_path = paste0(wf_dir, '/', post_image)
      post_ras <- terra::rast(post_path) %>%
```

```
                    terra::crop(terra::vect(fire %>% st_transform(4326)))
      post_ras = (post_ras-10000)/10000
    # Calculate dNBR & replace in pre-fire raster
      d_ras = pre_ras[[1]]- post_ras[[1]]
      pre_ras[[1]] <- d_ras
      names(pre_ras) <- c('dNBR', 'ndvi')
    # Sample rasters
      terra::extract(pre_ras[[1:2]],
                     pts %>%
                       terra::vect(),
                     fun = mean,
                     method = 'bilinear',
                     bind = TRUE)  %>%
      as_tibble() %>%
   add_column(FireCode=fire$FireCode, .before = 1 ) %>%
   bind_rows(., WfBurnIndices) -> WfBurnIndices
  }
```

Statistical analysis

Space-based vs. field-based data

```
PlotBurnIndices <- readxl::read_xlsx('SeverityComparisonData.xlsx',
                                     'PlotBurnIndices')
# Testing drivers of burn severity
  # Canopy temperature
    mc_0 <- lme4::lmer(dNBR ~ 1 + (1|location/burn),
                       REML = FALSE, data = PlotBurnIndices)
    mc_1 <- lme4::lmer(dNBR ~ MaxC + (1|location/burn),
                       REML = FALSE, data = PlotBurnIndices)
    BurnModels$Sev_v_MaxC <- anova(mc_0, mc_1)
  # Soil surface temperature
    sc_0 <- lme4::lmer(dNBR ~ 1 + (1|location/burn), REML = FALSE,
                       data = filter(PlotBurnIndices, !is.na(SoilMaxC) ) )
    sc_1 <- lme4::lmer(dNBR ~ SoilMaxC + (1|location/burn), REML = FALSE,
                       data = filter(PlotBurnIndices, !is.na(SoilMaxC)) )
    BurnModels$Sev_v_SoilMaxC <- anova(sc_0, sc_1)
  # Rate of spread
    rs_0 <- lme4::lmer(dNBR ~ 1 + (1|location/burn), REML = FALSE,
                        data = filter(PlotBurnIndices, ! is.na(ros)) )
    rs_1 <- lme4::lmer(dNBR ~ ros + (1|location/burn), REML = FALSE,
                        data = filter(PlotBurnIndices, ! is.na(ros)) )
    BurnModels$Sev_v_ROS <- anova(rs_0, rs_1)
```

Wildfires vs. Rx burns

```
reg_dat <- readxl::read_xlsx('SeverityComparisonData.xlsx',
                                     'BurnSeverityData') %>%
            rename(dNBR = dNBR_Mean) %>%
            select(zone, type, dNBR)

    sev_0 <- lm(dNBR ~ 1, data = reg_dat)
```

```r
sev_int <- lm(dNBR ~ type*zone, data = reg_dat)

anova(sev_0, sev_int)

BurnModels$type_comp <- emmeans::emmeans(sev_int, ~  type | zone)

emmeans::joint_tests(BurnModels$type_comp, by = 'zone')
```