04 - Climate Processing

University of Florida
Shelly Gaynor

iDigBio
Integrated Digitized Biocollections

Layer Processing

- Need layers in ASCII format
- Clip layers to fit your desired range
  - Dynamic alpha hull
    - + 80th quartile buffer
- Remove layers with collinearity based on variable inflation factor and permutation importance.
- Define (1) shared projection layers and (2) training layers.

03_ClimateProcessing.R

Load Packages

```
library(raster)
library(gtools)
library(sf)
library(rangeBuilder)
library(dplyr)
library(caret)
library(usdm)
library(dismo)
library(stringr)
```

Load Functions

Dr. Mike Belitz

```
source("functions/VIFLayerSelect.R")
```

- These functions are modified from mbelitz/Odo_SDM_Rproj (GitHub)

Load bioclim layers

```
biolist <- list.files("data/climate_processing/bioclim/", pattern =
"*.tif", full.names = TRUE)
```

Order list using gtools.

```
biolist <- mixedsort(sort(biolist))
```

Load rasters as a stack.

```
biostack <- raster::stack(biolist)
```

Load occurrence records

And fix the name column class and make sure it is a character.

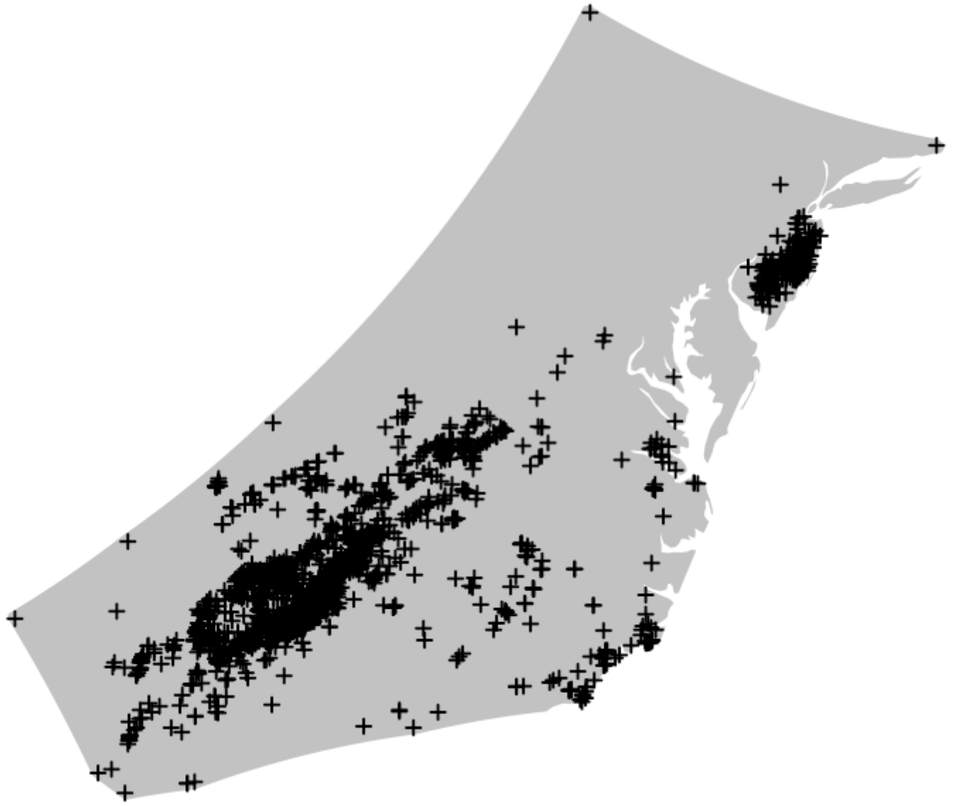
```
alldf <- read.csv("data/cleaning_demo/maxent_ready/diapensiaceae_maxentready_20230605.csv")
alldf$species <- as.character(alldf$species)
alldfsp <- st_as_sf(alldf, coords = c("longitude", "latitude"), crs = 4326)
```

Define extent

Create alpha hull

```
hull <- rangeBuilder::getDynamicAlphaHull(x = alldf,
                                          coordHeaders = c("longitude", "latitude"),
                                           fraction = 1, # min. fraction of records we want included
                                           partCount = 1, # number of polygons allowed
                                           initialAlpha = 20, # initial alpha size, 20m
                                           clipToCoast = "terrestrial",
                                           verbose = TRUE)
```

```
plot(hull[[1]], col=rangeBuilder::transparentColor('gray50', 0.5), border = NA)
points(x = alldf$longitude, y = alldf$latitude, cex = 0.5, pch = 3)
```



Define extent

Transform into CRS related to meters

```
hullTrans <- st_transform(hull[[1]], CRS("+proj=cea +lat_ts=0 +lon_0"))
alldfspTrans <- st_transform(alldfsp, CRS("+proj=cea +lat_ts=0 +lon_0"))
```

Calculate buffer size

Here we take the 80th quantile of the max distance between points

```
buffDist <- quantile(x = (apply(sf::st_distance(alldfspTrans), 2, FUN = function(x) sort(x)[2])),
                     probs = 0.80, na.rm = TRUE)
buffDist
```

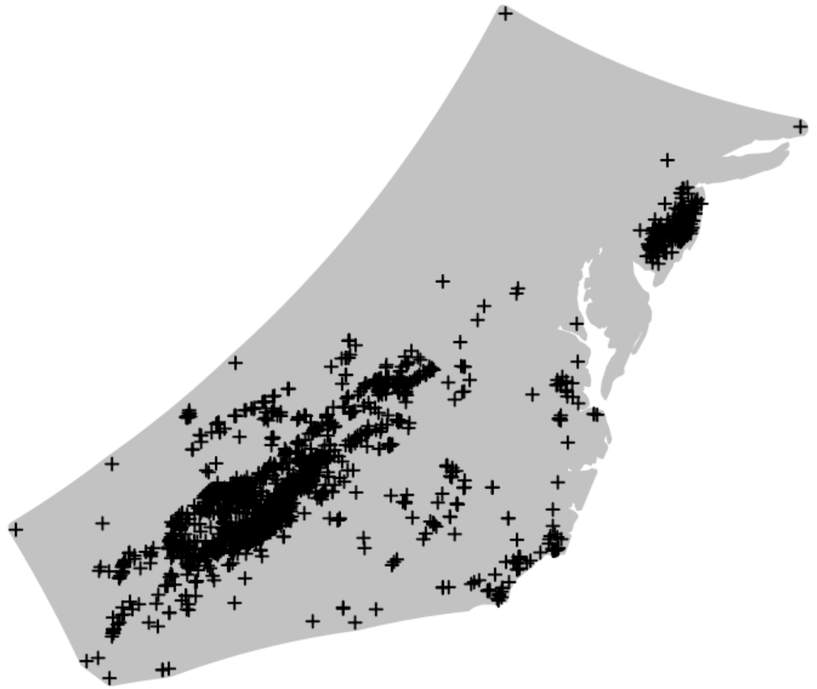
```
##      80%
## 3786.239
```

Buffer the hull

```
buffer_m <- st_buffer(x = hullTrans, dist = buffDist, dissolve = TRUE)
buffer <- st_transform(buffer_m,  CRS("+proj=longlat +datum=WGS84"))
```

Visualize

```
plot(buffer, col=rangeBuilder::transparentColor('gray50', 0.5), border = NA)
points(x = alldf$longitude, y = alldf$latitude, cex = 0.5, pch = 3)
```



Mask and Crop Layers

```
path <- "data/climate_processing/all/"
end <- ".asc"
for(i in 1:length(biolist)){
    # Subset raster layer
    rast <- biostack[[i]]
    # Setup file names
    name <- names(rast)
    out <- paste0(path, name)
    outfile <- paste0(out, end)
    # Crop and mask
    c <- crop(rast, sf::as_Spatial(buffer))
    c <- mask(c, sf::as_Spatial(buffer))
    # Write raster
    writeRaster(c, outfile, format = "ascii", overwrite = TRUE)
}
```

WARNING: TIME INTENSIVE

Select layers

We only want to include layers that are not highly correlated. To assess which layers we will include, we can look at the pearson correlation coefficient among layers. #### Stack all layers

```
clippedlist <- list.files("data/climate_processing/all/", pattern = "*.asc", full.names = TRUE)
clippedlist <- mixedsort(sort(clippedlist))
clippedstack <- raster::stack(clippedlist)
```

Then calculate the correlation coefficient

```
corr <- layerStats(clippedstack, 'pearson', na.rm=TRUE)
```

Isolate only the pearson correlation coefficient and take absolute value

```
c <- abs(corr$`pearson correlation coefficient`)
```

Write file and view in excel

```
write.csv(c, "data/climate_processing/correlationBioclim.csv", row.names = FALSE)
```

Highly correlated layers (> |0.80|) can impact the statistical significance of the niche models and therefore must be removed.

Select layers

Randomly select variables to remove

```
envtCor <- mixedsort(sort(findCorrelation(c, cutoff = 0.80, names = TRUE, exact = TRUE)))
envtCor
```

```
##  [1] "bio_1"  "bio_3"  "bio_4"  "bio_6"  "bio_9"  "bio_10" "bio_11" "bio_12"
##  [9] "bio_13" "bio_16" "bio_17" "bio_19"
```

WARNING: TIME INTENSIVE

Select layers

Variable inflation factor (VIF)

Warning: Time Intensive!
VIF can detect for multicollinearity in a set of multiple regression variables. Run a simple maxent model for every species and calculate the average permutation contribution.

Run preliminary MaxEnt models

Loop through each species and save permutation importance in list.
Warning: This will print warnings even when it works fine.

```
set.seed(195)
m <- c()
for(i in  1:length(unique(alldf$species))){
    name <- unique(alldf$species)[i]
    spp_df <-  alldf %>%
               dplyr::filter(species == name) %>%
               dplyr::select(longitude, latitude)
    model <- dismo::maxent(x = clippedstack, p = spp_df,
                           progress = "text", silent = FALSE)
    m[[i]] <- vimportance(model)
}
```

Select layers

Bind the dataframes

```
mc <- do.call(rbind, m)
```

Calculate the mean and rename columns

```
mc_average <- aggregate(mc[, 2], list(mc$Variables), mean)
mc_average <- mc_average %>%
              dplyr::select(Variables = Group.1, permutation.importance = x)
mc1 <- mc_average
```

Select layers

Use VIF and the MaxEnt permutation importance to select the best variables for your model. Note, this leads to different layers when the models are rerun without setting seed due to permutations being random.

```
selectedlayers <- VIF_layerselect(clippedstack, mc_average)
mixedsort(sort(names(selectedlayers)))
```

Correct set for workshop purposes

Since this can vary per system (despite setting seed), we added this line to keep our files consistent for the workshop

```
sl <- c("bio_3", "bio_7", "bio_8", "bio_9", "bio_14", "bio_15", "bio_18", "elev")
selectedlayers <- raster::subset(clippedstack, sl)
```

Select layers

Move selected layers to “Present_Layers/all/” for use in MaxE later.

```
for(i in 1:length(names(selectedlayers))){
    name <- names(selectedlayers)[i]
    from <- paste0("data/climate_processing/all/", name, ".asc")
    to <- paste0("data/climate_processing/PresentLayers/all/", name, ".asc")
    file.copy(from, to,
              overwrite = TRUE, recursive = FALSE,
              copy.mode = TRUE)
}
```

WARNING: TIME INTENSIVE

Create Species Training Layers

```
for(i in 1:length(unique(alldf$species))){
    sname <- unique(alldf$species)[i]
    # Subset species from data frame
    spp_df <-  alldf %>%
               dplyr::filter(species == sname)
    spp_dfsp <- st_as_sf(spp_df, coords = c("longitude", "latitude"), crs = 4326)
    ## Create alpha hull
    sphull <- rangeBuilder::getDynamicAlphaHull(x = spp_df,
                                                coordHeaders = c("longitude", "latitude"),
                                                fraction = 1, # min. fraction of records we want included
                                                partCount = 1, # number of polygons allowed
                                                initialAlpha = 20, # initial alpha size, 20m
                                                clipToCoast = "terrestrial",
                                                verbose = TRUE)
    ## Add buffer to hull
    ### Transform into CRS related to meters with Equal Area Cylindrical projection
    sphullTrans <- st_transform(sphull[[1]], CRS("+proj=cea +lat_ts=0 +lon_0"))
    spp_dfTrans <- st_transform(spp_dfsp, CRS("+proj=cea +lat_ts=0 +lon_0"))

    ### Calculate buffer size
    #### Here we take the 80th quantile of the max distance between points
    spbuffDist <- quantile(x = (apply(sf::st_distance(spp_dfTrans), 2, FUN = function(x) sort(x)[2])),
                           probs = 0.80, na.rm = TRUE)
```

WARNING: TIME INTENSIVE

Create Species Training Layers

```
    ### Buffer the hull
    spbuffer_m <- st_buffer(x = sphullTrans, dist = spbuffDist, dissolve = TRUE)
    spbuffer <- st_transform(spbuffer_m,  CRS("+proj=longlat +datum=WGS84"))

    ### Crop and Mask
    spec <- gsub(" ", "_", sname)
    path <- paste0("data/climate_processing/PresentLayers/", spec,"/")
    end <- ".asc"
    for(j in 1:length(names(selectedlayers))){
        # Subset raster layer
        rast <- selectedlayers[[j]]
        # Setup file names
        name <- names(rast)
        out <- paste0(path, name)
        outfile <- paste0(out, end)
        # Crop and mask
        c <- crop(rast, sf::as_Spatial(spbuffer))
        c <- mask(c, sf::as_Spatial(spbuffer))
        # Write raster
        writeRaster(c, outfile, format = "ascii", overwrite = TRUE)
    }
}
```