240711_psk_use_pa_sampling_all

Pranav K.

```r
#devtools::install_github("danddr/USE")

library(USE)
library(terra)
library(raster)
library(sf)
#library(tidyverse)
```

Guanarito virus - *Z.brevicauda* and *S.alstoni*

```r
envdata <- do.call(brick, lapply(list.files(
  path = "./Data/Input/Processed/Resampled/guan",
  pattern = "*.tif", full.names = T),raster))

e <- extent(envdata)
zyg_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_zyg_brev01.csv")
process_gbif <- function(presence_data, e, name_longitude, name_latitude, date){
  gbif <- data.frame(lon = rep(NA, nrow(presence_data),
                               lat = rep(NA, nrow(presence_data))))
  gbif$lon <- presence_data[,name_longitude]
  gbif$lat <- presence_data[,name_latitude]
  gbif <- unique(gbif)
  gbif <- gbif[which(gbif$lon>=e[1] & gbif$lon<=e[2]),]
  gbif <- gbif[which(gbif$lat>=e[3] & gbif$lat<=e[4]),]
  return(gbif)
}
zyg_pres <- zyg_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

zyg_pres <- process_gbif(zyg_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(zyg_pres, coords=c("lon", "lat"), crs=4326)


myGrid.psAbs <- USE::paSampling(env.rast=envdata,
                                pres=myPres,
                                thres=0.75,
                                H=NULL,
                                grid.res=1,
                                n.tr = as.numeric(nrow(myPres)),
                                prev=NULL,
                                sub.ts=F,
                                n.ts=NULL,
                                plot_proc=F,
                                verbose=T)
save(myGrid.psAbs, file = "./Data/Input/Processed/paSampling/zyg.Rdata")
rm(list = setdiff(ls(), c("envdata", "process_gbif", "e")))
```

```
sig_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_sig01.csv")
sig_pres <- sig_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

sig_pres <- process_gbif(sig_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(sig_pres, coords=c("lon", "lat"), crs=4326)

sig.psAbs <-USE::paSampling(env.rast=envdata,
                            pres=myPres,
                            thres=0.75,
                            H=NULL,
                            grid.res=1,
                            n.tr = as.numeric(nrow(myPres)),
                            prev=NULL,
                            sub.ts=F,
                            n.ts=NULL,
                            plot_proc=F,
                            verbose=T)
save(sig.psAbs, file = "./Data/Input/Processed/paSampling/sig.Rdata")
rm(list = setdiff(ls(), "process_gbif"))
```

Machupo virus: *C.callosus*

```
envdata <- do.call(brick, lapply(list.files(
  path = "./Data/Input/Processed/Resampled/machu",
  pattern = "*.tif", full.names = T),raster))
e <- extent(envdata)

ccal_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_ccal01.csv")
ccal_pres <- ccal_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

ccal_pres <- process_gbif(ccal_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(ccal_pres, coords=c("lon", "lat"), crs=4326)
ccal.psAbs <-USE::paSampling(env.rast=envdata,
                             pres=myPres,
                             thres=0.75,
                             H=NULL,
                             grid.res=1,
                             n.tr = as.numeric(nrow(myPres)),
                             prev=NULL,
                             sub.ts=F,
                             n.ts=NULL,
                             plot_proc=F,
                             verbose=T)
```

```
save(ccal.psAbs, file = "./Data/Input/Processed/paSampling/ccal.Rdata")
rm(list = setdiff(ls(), "process_gbif"))
```

Junin virus: *C.musculinus*, *C.laucha* and *O.flavescens*

```
envdata <- do.call(brick, lapply(list.files(
  path = "./Data/Input/Processed/Resampled/junin",
  pattern = "*.tif", full.names = T),raster))
e <- extent(envdata)

cmus_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_cmus01.csv")
cmus_pres <- cmus_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

cmus_pres <- process_gbif(cmus_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(cmus_pres, coords=c("lon", "lat"), crs=4326)
cmus.psAbs <-USE::paSampling(env.rast=envdata,
                             pres=myPres,
                             thres=0.75,
                             H=NULL,
                             grid.res=1,
                             n.tr = as.numeric(nrow(myPres)),
                             prev=NULL,
                             sub.ts=F,
                             n.ts=NULL,
                             plot_proc=F,
                             verbose=T)
save(cmus.psAbs, file = "./Data/Input/Processed/paSampling/cmus.Rdata")
rm(list = setdiff(ls(), c("process_gbif", "envdata", "e")))
```

```
cla_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_cla01.csv")
cla_pres <- cla_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

cla_pres <- process_gbif(cla_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(cla_pres, coords=c("lon", "lat"), crs=4326)
cla.psAbs <-USE::paSampling(env.rast=envdata,
                            pres=myPres,
                            thres=0.75,
                            H=NULL,
                            grid.res=1,
                            n.tr = as.numeric(nrow(myPres)),
                            prev=NULL,
                            sub.ts=F,
                            n.ts=NULL,
                            plot_proc=F,
                            verbose=T)
save(cla.psAbs, file = "./Data/Input/Processed/paSampling/cla.Rdata")
rm(list = setdiff(ls(), c("process_gbif", "envdata", "e")))
```

```
ofl_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_ofl01.csv")
ofl_pres <- ofl_pres %>% dplyr::select(lon, lat) %>% dplyr::mutate(CLASS = rep(1, nrow(.)))

ofl_pres <- process_gbif(ofl_pres, e = e, "lon", "lat", NA)

myPres <- st_as_sf(ofl_pres, coords=c("lon", "lat"), crs=4326)
ofl.psAbs <-USE::paSampling(env.rast=envdata,
                            pres=myPres,
                            thres=0.75,
                            H=NULL,
                            grid.res=1,
                            n.tr = as.numeric(nrow(myPres)),
                            prev=NULL,
                            sub.ts=F,
                            n.ts=NULL,
                            plot_proc=F,
                            verbose=T)
save(ofl.psAbs, file = "./Data/Input/Processed/paSampling/ofl.Rdata")

rm(list = ls())
```

Reload all saved data

```
load("./Data/Input/Processed/paSampling/zyg.Rdata")
zyg.psABs <- myGrid.psAbs
rm(myGrid.psAbs)

load("./Data/Input/Processed/paSampling/sig.Rdata")
load("./Data/Input/Processed/paSampling/ccal.Rdata")
load("./Data/Input/Processed/paSampling/cmus.Rdata")
load("./Data/Input/Processed/paSampling/cla.Rdata")
load("./Data/Input/Processed/paSampling/ofl.Rdata")
```

functions to process presence-absence

```
process_gbif <- function(presence_data, e, name_longitude, name_latitude, date){
  gbif <- data.frame(lon = rep(NA, nrow(presence_data),
                               lat = rep(NA, nrow(presence_data))))
  gbif$lon <- presence_data[,name_longitude]
  gbif$lat <- presence_data[,name_latitude]
  gbif <- unique(gbif)
  gbif <- gbif[which(gbif$lon>=e[1] & gbif$lon<=e[2]),]
  gbif <- gbif[which(gbif$lat>=e[3] & gbif$lat<=e[4]),]
  return(gbif)
}
process_absences <- function(abs_df, pres_df, e_df, coord_ref,
                             path_write, layer_name, driver_write) {
  abs_df <- abs_df %>% select(lon = x,lat = y)
  abs_df <- as.data.frame(abs_df[,1:2])
  abs_df$geometry <- NULL
  pres_df <- pres_df %>% select("lon", "lat")
  pres_df <- process_gbif(pres_df, e = e_df, "lon", "lat", NA)
  train <- rbind(pres_df, abs_df)
  pa_train <- c(rep(1, nrow(pres_df)), rep(0, nrow(abs_df)))
  train <- data.frame(cbind(CLASS=pa_train, train))
  crs <- crs(coord_ref)
  train <- train[sample(nrow(train)),]
  class.pa <- data.frame(train[,1])
  colnames(class.pa) <- 'CLASS'
  dataMap.gbif  <- SpatialPointsDataFrame(train[,c(2,3)], class.pa,
                                          proj4string =crs)
  st_write(as(dataMap.gbif, "sf"),
           path_write,layer_name,
           driver = driver_write, append = F)
  return(dataMap.gbif)
}
```

reference rasters, crs, extent

```
env_guan <- raster("./Data/Input/Processed/Resampled/guan/bclim_01.tif")
env_machu <- raster("./Data/Input/Processed/Resampled/machu/bclim_01.tif")
env_junin <- raster("./Data/Input/Processed/Resampled/junin/bclim_01.tif")

e_guan <- extent(env_guan)
e_machu = extent(env_machu)
e_junin = extent(env_junin)
```

Load presence-only data

```
zyg_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_zyg_brev01.csv")
sig_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_sig01.csv")
ccal_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_ccal01.csv")
cmus_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_cmus01.csv")
cla_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_cla01.csv")
ofl_pres <- read.csv("./Data/Input/Raw/GBIF/gbif_ofl01.csv")
```

final processing and save shapefiles

```
zyg_gbif <- process_absences(abs_df = zyg.psABs,
                             pres_df = zyg_pres,
                             e_df = e_guan,
                             coord_ref = env_guan,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "zyg_pa",
                             driver_write = "ESRI Shapefile")
```

```r
ggplot(st_as_sf(zyg_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()

sig_gbif <- process_absences(abs_df = sig.psAbs,
                             pres_df = sig_pres,
                             e_df = e_guan,
                             coord_ref = env_guan,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "sig_pa",
                             driver_write = "ESRI Shapefile")

ggplot(st_as_sf(sig_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()

ccal_gbif <- process_absences(abs_df = ccal.psAbs,
                             pres_df = ccal_pres,
                             e_df = e_machu,
                             coord_ref = env_machu,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "ccal_pa",
                             driver_write = "ESRI Shapefile")

ggplot(st_as_sf(ccal_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()

cmus_gbif <- process_absences(abs_df = cmus.psAbs,
                             pres_df = cmus_pres,
                             e_df = e_junin,
                             coord_ref = env_junin,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "cmus_pa",
                             driver_write = "ESRI Shapefile")

ggplot(st_as_sf(cmus_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()

cla_gbif <- process_absences(abs_df = cla.psAbs,
                             pres_df = cla_pres,
                             e_df = e_junin,
                             coord_ref = env_junin,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "cla_pa",
                             driver_write = "ESRI Shapefile")

ggplot(st_as_sf(cla_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()

ofl_gbif <- process_absences(abs_df = ofl.psAbs,
                             pres_df = ofl_pres,
                             e_df = e_junin,
                             coord_ref = env_junin,
                             path_write = "./Data/Input/Processed/GBIF/paSampling",
                             layer_name = "ofl_pa",
                             driver_write = "ESRI Shapefile")

ggplot(st_as_sf(ofl_gbif), aes(color = factor(CLASS))) + geom_sf() + theme_linedraw()
```