dev\_blocks\_observed\_in

N.M. Tarr

2023-10-2

# Purpose

This document details a function that returns a data frame of NCBA blocks within which a species was observed. Parameter are available that limit the results according to whether the species was observed within, or outside of, a specified time period (eg., breeding dates). Furthermore, output can be limited to certain breeding categories.

# Definition

blocks\_observed\_in

## function (observations, start\_day = 0, end\_day = 366, within = TRUE,   
## breeding\_categories = c("C4", "C3", "C2", "C1", ""))   
## {  
## if (within == TRUE) {  
## obs <- observations %>% filter(yday(observation\_date) >=   
## start\_day & yday(observation\_date) <= end\_day)  
## }  
## if (within == FALSE) {  
## obs <- observations %>% filter(yday(observation\_date) <   
## start\_day | yday(observation\_date) > end\_day)  
## }  
## obs <- filter(obs, breeding\_category %in% breeding\_categories)  
## fields <- c("atlas\_block", "common\_name")  
## obs <- obs %>% select(fields) %>% distinct()  
## return(obs)  
## }

# Usage

This section demonstrates how to apply this function to map blocks with different type sof observations for a species. The first steps are to get the breeding safe dates as day of the year from the AtlasCache, as well as a data frame of observations. The observation data frame should be converted to the EBD format and records with an observation count of 0 should be removed.

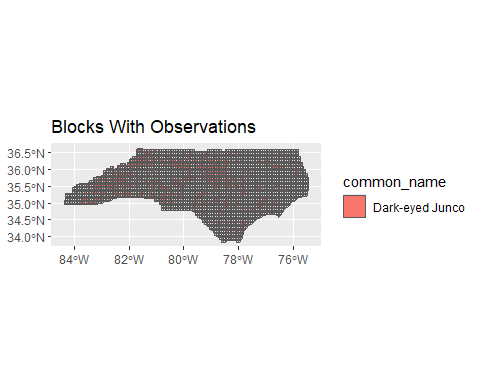
# Species  
species <- "Dark-eyed Junco"  
  
# Pull out breeding season records  
breedates <- get\_breeding\_dates(species, day\_year = TRUE)  
  
# Get all the observations for the species, exclude zero count records though  
observations <- get\_observations(species = species,   
 database = "AtlasCache",   
 fields = NULL) %>%  
 to\_EBD\_format() %>%  
 filter(observation\_count != 0)

Next, retrieve a spatially-enabled blocks data frame.

# Get a blocks data frame with simple features  
fields <- c("ID\_BLOCK\_CODE", "ID\_EBD\_NAME")  
blocks\_sf <- get\_blocks(spatial = TRUE, fields = fields)

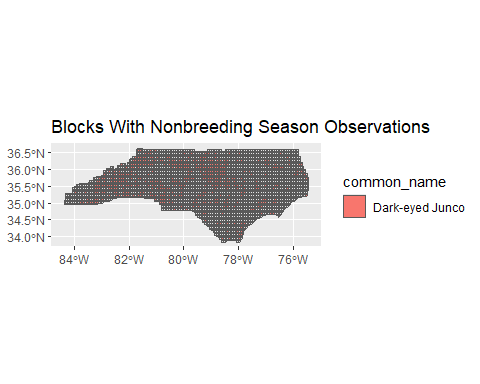
The function can then be used to get data frames of blocks with records for the species. Those data frames can then be joined with the blocks data frame to create a map.

# Get blocks with any type of record from any day of the year.  
pres\_blocks <- blocks\_observed\_in(observations, start\_day = 0, end\_day = 366,   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
# Join with blocks data frame  
pres\_blocks\_sf <- right\_join(blocks\_sf, pres\_blocks, by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = pres\_blocks\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks With Observations")



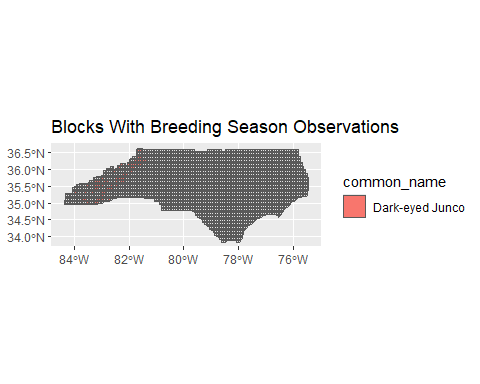
Map blocks with nonbreeding season observations.

winter\_blocks <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = FALSE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
  
# Join with blocks data frame  
winter\_blocks\_sf <- right\_join(blocks\_sf, winter\_blocks,   
 by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = winter\_blocks\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks With Nonbreeding Season Observations")



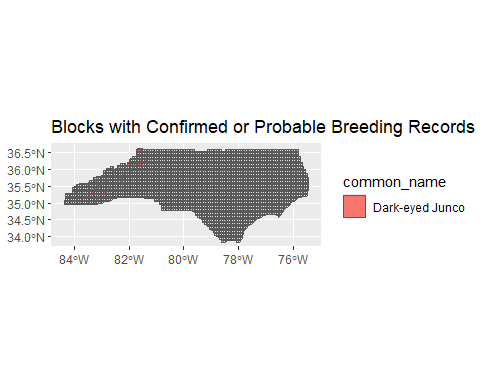
Map the blocks with breeding season observations.

summer\_blocks <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
# Join with blocks data frame  
summer\_blocks\_sf <- right\_join(blocks\_sf, summer\_blocks,   
 by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = summer\_blocks\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks With Breeding Season Observations")



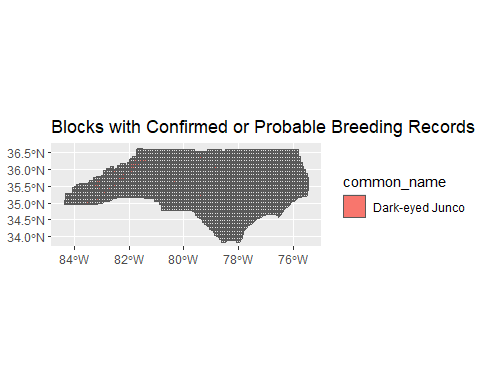
Map blocks with confirmed or probable breeding observations from within breeding safe dates.

conf\_prob <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = TRUE,  
 breeding\_categories = c("C3", "C4"))  
  
conf\_prob\_sf <- right\_join(blocks\_sf, conf\_prob,   
 by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = conf\_prob\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks with Confirmed or Probable Breeding Records Within Safe Dates")



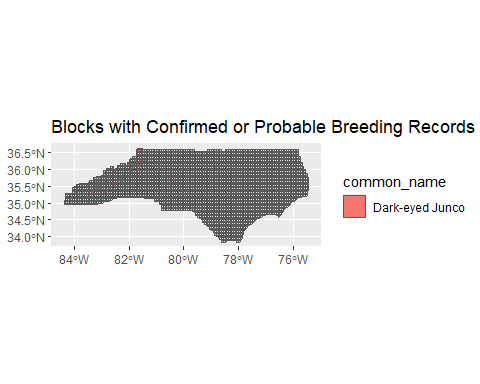
Map blocks with confirmed or probable breeding observations from outside of breeding safe dates.

conf\_prob\_out <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = FALSE,  
 breeding\_categories = c("C3", "C4"))  
  
conf\_prob\_out\_sf <- right\_join(blocks\_sf, conf\_prob\_out,   
 by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = conf\_prob\_out\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks with Confirmed or Probable Breeding Records Outside Safe Dates")



Map blocks with confirmed or probable breeding observations from any day of the year.

confirmed <- blocks\_observed\_in(observations, start\_day = 0,   
 end\_day = 366,   
 within = TRUE,  
 breeding\_categories = c("C3", "C4"))  
  
# Join with blocks data frame  
confirmed\_sf <- right\_join(blocks\_sf, confirmed,   
 by = join\_by("ID\_BLOCK\_CODE" == "atlas\_block"))  
  
# Plot the spatial data frame  
ggplot() +  
 geom\_sf(data = blocks\_sf) +  
 geom\_sf(data = confirmed\_sf, aes(fill = common\_name)) +  
 ggtitle("Blocks with Confirmed or Probable Breeding Records")



# Tests

Test that start\_day and end\_day work by comparing three results: a data frame of blocks with an observation on any day, a data frame of blocks with observations from within a time period, and a data frame of blocks with observations from outside of a time period. Also, a list of all atlas\_blocks that are present in the observations data frame is useful.

# Get blocks with any type of record from any day of the year.  
any <- blocks\_observed\_in(observations, start\_day = 0, end\_day = 366,   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
  
# Get blocks with any type of record from between the 100 and 200th days of the year.  
within\_ <- blocks\_observed\_in(observations, start\_day = 100, end\_day = 200,   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
  
# Get blocks with any type of record from between the 100 and 200th days of the year.  
outside <- blocks\_observed\_in(observations, start\_day = 100, end\_day = 200,   
 within = FALSE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
  
# Get a list of blocks from observations  
obs\_blocks <- unique(observations$atlas\_block)

The list of unique blocks in observations should be the same as the list from the any data frame above. This test is passed if the chunck below returns TRUE.

print(length(setdiff(obs\_blocks, any$atlas\_block)) == 0)

## [1] TRUE

The within data frame should be a subset of the any data frame. This test is passed if the chunk returns TRUE.

# Is within\_ a subset of any?  
print(length(setdiff(within\_$atlas\_block, any$atlas\_block)) == 0)

## [1] TRUE

The outside data frame should be a subset of the any data frame. This test is passed if the chunk returns TRUE.

# Is outside a subset of any?  
print(length(setdiff(outside$atlas\_block, any$atlas\_block)) == 0)

## [1] TRUE

All atlas\_blocks from any should be in either the within or outside data frames. This test is passed if the chunk only returns TRUE.

# Get the union of the within and outside lists  
u <- union(outside$atlas\_block, within\_$atlas\_block)  
  
# Test that is has the same items as the any list  
print(length(setdiff(u, any$atlas\_block)) == 0)

## [1] TRUE

print(length(setdiff(any$atlas\_block, u)) == 0)

## [1] TRUE

Asking for blocks with confirmed breeding observations within the safe dates should return a subset of the blocks with any category of observation within the safe dates.

# Get two results where the only difference is the categories allowed:1) all  
# codes and 2) just confirme  
in\_safe <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2",   
 "C1", ""))  
  
confirmed\_in\_safe <- blocks\_observed\_in(observations, start\_day = breedates[1],   
 end\_day = breedates[2],   
 within = TRUE,  
 breeding\_categories = c("C4"))  
  
# Test that confirmed in safe is a subset of in\_safe  
print(length(setdiff(confirmed\_in\_safe$atlas\_block, in\_safe$atlas\_block)) == 0)

## [1] TRUE

# Speed

Run the function 10 times and record the runtime

time <- c()  
for (i in 1:10) {  
 time1 <- proc.time()  
 blocks\_observed\_in(observations, start\_day = 100, end\_day = 200,   
 within = TRUE,  
 breeding\_categories = c("C4", "C3", "C2"))  
 t <- proc.time() - time1  
 time[i] <- t["elapsed"]  
}  
  
# Print the descriptive statistics  
print(summary(time))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0600 0.0625 0.0750 0.1360 0.1050 0.5100