Thanks to the [second post of the series where we obtained data from  
eBird](https://ropensci.org/blog/2018/08/21/birds-radolfzell/) we know  
what birds were observed in the county of Constance. Now, not all  
species’ names mean a lot to me, and even if they did, there are a lot  
of them. In this post, we shall use rOpenSci’s packages accessing  
taxonomy and trait data in order to summarize some characteristics of  
the birds’ population of the county: armed with scientific and common  
names of birds, we have access to plenty of open data!

**Getting and filtering the occurrence data**

For more details about the following code, refer the single difference is our adding a step to keep only data for the  
most recent years.

# polygon for filtering

landkreis\_konstanz <- osmdata::getbb("Landkreis Konstanz",

format\_out = "sf\_polygon")

crs <- sf::st\_crs(landkreis\_konstanz)

# get and filter data

f\_out\_ebd <- "ebird/ebd\_lk\_konstanz.txt"

library("magrittr")

ebd <- auk::read\_ebd(f\_out\_ebd) %>%

sf::st\_as\_sf(coords = c("longitude", "latitude"),

crs = crs)

in\_indices <- sf::st\_within(ebd, landkreis\_konstanz)

ebd <- dplyr::filter(ebd, lengths(in\_indices) > 0)

ebd <- as.data.frame(ebd)

ebd <- dplyr::filter(ebd, approved, lubridate::year(observation\_date) > 2010)

nrow(ebd)

## [1] 8599

We will also need these two data.frames later: abundance by species, and  
dictionary of names.

abundance <- dplyr::count(ebd, scientific\_name)

dico <- unique(dplyr::select(ebd, scientific\_name,

common\_name))

**Getting taxonomic information**

In this section I would like to get an idea of how diverse the types of  
birds are in the County of Constance. I want to draw a phylogenetic tree  
of the local species, and for that, I’ll first retrieve the  
classification for each species from  
[NCBI](https://www.ncbi.nlm.nih.gov/taxonomy) using the taxize package that “allows users to  
search over many taxonomic data sources for species names (scientific  
and common) and download up and downstream taxonomic hierarchical  
information – among other things.”.

We first query \*uid\*’s and then use the classification function,  
instead of passing the species name directly to classification,  
because the IDs are unique whereas results for species names aren’t.  
Rate limiting is thankfully managed by the package itself so we users do  
not need to worry about that.

Library(Taxize)

ids <- taxize::get\_uid(unique(ebd$scientific\_name))

classif <- taxize::classification(ids)

fs::dir\_create("taxo")

save(classif, file = file.path("taxo", "classif.RData"))

There are 211 species, we get 211 elements in the classification  
(sum(lengths(classif)==3)), that is a list of data.frames:

load(file.path("taxo", "classif.RData"))

str(classif[[1]])

## 'data.frame': 31 obs. of 3 variables:

## $ name: chr "cellular organisms" "Eukaryota" "Opisthokonta" "Metazoa" ...

## $ rank: chr "no rank" "superkingdom" "no rank" "kingdom" ...

## $ id : chr "131567" "2759" "33154" "33208" ...

Now, we’ll represent the whole taxonomy as a tree, using the handy  
taxize::class2tree function and the great ggtree.

Library(ggtree)

tree <- taxize::class2tree(classif)

library("ggplot2")

ggtree::ggtree(tree$phylo) +

ggtree::geom\_tiplab(aes(), size = 2, vjust=0.25) +

xlim(0, 150)

labelled tree of all
species

This tree is… unreadable. But at this point, it’s worth remembering that  
we got here using three taxize functions only: taxize::get\_uid,  
taxize::classification and taxize::class2tree. What a smooth  
workflow!

There are 18 orders and I do not intend to add the highlighting command  
for each of them by hand! I’ll streamline the process, starting by  
automatically extracting the node ID of each order. The solution below  
might be a little over-complicated, so R taxonomy experts, please chime  
in! I transformed the tree phylo object to a phylo4 from the  
phylobase package, in order to easily retrieve all  
ancestor nodes for any group of species. Within an order, the order node  
ID is the highest common node ID.

Library(Phylobase)

p4 <- phylobase::phylo4(tree$phylo)

# helper to translate labels

translate <- function(scientific\_name){

if(scientific\_name %in% dico$scientific\_name){

dico$common\_name[dico$scientific\_name == scientific\_name]

}else{

scientific\_name

}

}

find\_order\_node\_id <- function(order, p4){

order\_members <- as.character(tree$classification$species[tree$classification$order == order])

nodes <- phylobase::ancestors(p4, order\_members, type = "ALL")

# the ID is the higher common node IDs

nodes <- purrr::map(nodes, as.numeric)

if(length(order\_members) > 1){

id <- max(Reduce(intersect, nodes))

}else{

id <- min(unlist(nodes))

}

common\_names <- purrr::map\_chr(order\_members, translate)

species <- stringr::str\_wrap(toString(common\_names),width = 50)

tibble::tibble(id = id,

order = order,

size = length(order\_members),

species = species)

}

orders <- purrr::map\_df(unique(tree$classification$order),

find\_order\_node\_id,

p4)

str(orders)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 18 obs. of 4 variables:

## $ id : num 237 238 235 239 231 10 228 226 224 222 ...

## $ order : Factor w/ 18 levels "Anseriformes",..: 3 12 13 1 5 4 15 11 16 17 ...

## $ size : int 32 86 9 35 5 1 7 6 5 3 ...

## $ species: chr "Black-headed Gull, Common Tern, Northern Lapwing,\nYellow-legged Gull, Common Sandpiper, Eurasian\nCurlew, Mew "| \_\_truncated\_\_ "Carrion Crow, Eurasian Magpie, House Sparrow,\nShort-toed Treecreeper, Eurasian Blackbird,\nEuropean Greenfinch"| \_\_truncated\_\_ "Gray Heron, Great Cormorant, Great Egret, Eurasian\nSpoonbill, Purple Heron, Cattle Egret, Little\nBittern, Lit"| \_\_truncated\_\_ "Mallard, Mute Swan, Common Goldeneye, Common\nMerganser, Common Pochard, Ferruginous Duck,\nGreen-winged Teal, "| \_\_truncated\_\_ ...

For each order, I’ll get a silhouette from Phylopic using rphylopic Package.

Library(rphylopic)

get\_results <- function(name){

id <- rphylopic::name\_search(name)

rphylopic::name\_images(id$canonicalName[1,1])

}

get\_pic <- function(order, classification){

message(order)

# shortcurt for flamingos

if(order == "Phoenicopteriformes"){

return(tibble::tibble(pic\_id = "28473411-c079-4654-bbb7-34a5615bb414",

order = "Phoenicopteriformes"))

}

classification <- classification[classification$order == order,]

results <- get\_results(order)

if (length(results$same) > 0){

# best case

pic\_id <- results$`same`[[1]]$`uid`

}else{

# take the most common species

# and get any pic of it

results <- get\_results(classification$species[

classification$n == max(classification$n, na.rm=TRUE)

][1])

results <- purrr::keep(results, function(x) length(x) > 0)

results <- unlist(results)

results <- results[length(results)]

pic\_id <- as.character(results)

}

tibble::tibble(pic\_id = pic\_id,

order = order)

}

library("magrittr")

classification <- tree$classification %>%

dplyr::mutate(species = as.character(species)) %>%

dplyr::left\_join(abundance,

by = c("species" = "scientific\_name"))

ids <- purrr::map\_df(orders$order, get\_pic, classification)

save(ids, file = file.path("taxo", "ids.RData"))

It is rather tricky to automatically get pics from Phylopic since you  
might not get one for the order itself, maybe one for the subtaxon  
instead, etc, so we made decisions blindly in the script above. In real  
life one might prefer selecting IDs by hand.

Now, we can highlight each order!

Library(ggtree)

# Get pics ids

load(file.path("taxo", "ids.RData"))

# Plot basic tree

p <- ggtree::ggtree(tree$phylo)

# Sort the orders by node id

orders <- dplyr::arrange(orders, - id)

# Helper to plot one order

plot\_order <- function(order, orders,

ids, p){

# Get index

i <- which(orders$order == order)

# From image ID get image itself

# and image metadata (copyright &co)

img\_id <- ids$pic\_id[ids$order == order]

img <- rphylopic::image\_data(img\_id, 512)

img\_info <- rphylopic::image\_get(img\_id,

options = c("credit",

"licenseURL"))

if(is.null(img\_info$credit)){

img\_info$credit <- ""

}

# Now, plot!

p +

# Highlight the order

ggtree::geom\_hilight(node = orders$id[i],

fill = "salmon") +

# Order name as title

ggtitle(orders$order[i])+

xlim(0, 150) +

ylim(0, 250) +

# Add species names on the side

annotate("text", x = 110,

y = 200, label = orders$species[i],

size = 4) +

# Credit at the bottom

annotate("text", x = 110,

y = 0,

size = 2,

label = glue::glue("Silhouette: {img\_info$credit}\n{img\_info$licenseURL}"))

# Save a first time

filepath <- file.path("taxo", glue::glue("p{i}.png"))

ggsave(filepath, width = 7, height = 7)

# Add silhouette via magick

silhouette <- magick::image\_read(img[[1]])

magick::image\_read(filepath) %>%

magick::image\_composite(silhouette,

offset = "+1300+1400") %>%

magick::image\_write(filepath)

}

# Create aaall plots

purrr::walk(orders$order, plot\_order,

orders, ids, p)

Once we have created all these PNGs, we can join them into a gif using install.packages("gifski")

png\_files <- fs::dir\_ls("taxo", regexp = "[.]png$")

gifski::gifski(png\_files = png\_files,

gif\_file = file.path("2018-09-04-birds-taxo-traits\_files",

"figure-markdown\_github", "taxo.gif"),

delay = 3,

width = 500, height = 500,

progress = FALSE)

## [1] "/img/blog-images/2018-09-04-birds-taxo-traits/taxo.gif"

knitr::include\_graphics(file.path("2018-09-04-birds-taxo-traits\_files",

"figure-markdown\_github", "taxo.gif"))

animated tree with species names and order
silhouette

This gif shows many species names and  
[orders](https://en.wikipedia.org/wiki/Category:Bird_orders) giving us a  
feeling for what we might encounter in the county of Constance, but it  
lacks quantitative information about the species. It’d be interesting to  
create trees such as the ones of the metacoder package to reflect abundance,  
possibly depending on the very local area (distance to watery area) or  
season, potentially using the taxize::downstream function to get all  
families in each order, even families not present in our occurrence  
dataset. This idea is beyond the scope of this post. What *is* in scope,  
now, is trying to get trait information for the species.

**Getting trait information**

In ecology, *traits* are characteristics of organisms such as habitat,  
body size, threats, etc. It’s a whole bunch of data you can get for free  
based on species scientific names, from different data providers. The traits package, part of the  
rOpenSci’s suite, is an interface to various sources of traits data. In  
this section, we shall use data from BirdLife International: habitat and  
threats.

The different functions of traits have prefixes indicating with which  
data source they interact. Here we shall use traits::birdlife\_habitat  
and traits::birdlife\_threats. To get access to the data available for  
each species, we first need to get its IUCN ID using the taxize package (or the rredlist package that it wraps)

species <- unique(ebd$scientific\_name)

get\_info <- function(species){

message(species)

Sys.sleep(1)

iucn\_id <- taxize::iucn\_id(species)

if(!is.na(iucn\_id)){

habitat <- traits::birdlife\_habitat(iucn\_id)

threats <- traits::birdlife\_threats(iucn\_id)

}else{

habitat <- NULL

threats <- NULL

}

list(habitat = habitat,

threats = threats)

}

species\_info <- purrr::map(species,

get\_info)

names(species\_info) <- species

save(species\_info, file = file.path("taxo", "species\_info.RData"))

The script above isn’t the smartest since it doesn’t retain information  
about the species names. Not too bad here because I want to get a  
general idea of birds’ habitats and threats over the county.

habitat <- purrr::map\_df(species\_info, "habitat")

Out of 211, 203 are represented in this dataset.

library("ggplot2")

habitat %>%

janitor::clean\_names() %>%

dplyr::group\_by(id) %>%

dplyr::mutate(ok = all(c("breeding", "non-breeding") %in% occurrence)) %>%

dplyr::ungroup() %>%

dplyr::filter(ok, importance == "major") %>%

dplyr::mutate(habitat = dplyr::case\_when(stringr::str\_detect(habitat\_level\_1,

"Marine") ~ "Marine",

stringr::str\_detect(habitat\_level\_1,

"Rocky") ~ "Rocky",

TRUE ~ habitat\_level\_1)) %>%

ggplot() +

geom\_bar(aes(occurrence, fill = habitat),

position = "fill") +

# palette recommended in https://github.com/clauswilke/colorblindr

# for people with color-vision deficiency

colorblindr::scale\_fill\_OkabeIto() +

theme(legend.position = "bottom") +

hrbrthemes::theme\_ipsum(base\_size = 12,

axis\_title\_size = 12,

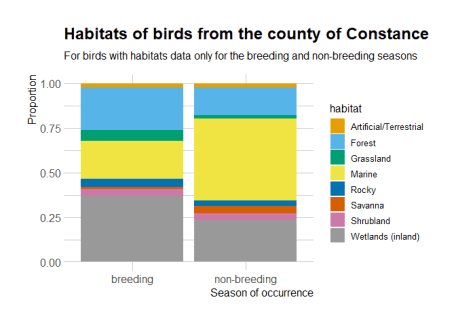
axis\_text\_size = 12) +

ggtitle("Habitats of birds from the county of Constance",

subtitle = "For birds with habitats data only for the breeding and non-breeding seasons") +

ylab("Proportion") +

xlab("Season of occurrence")



It seems that the birds present in the County of Constance might have  
different distributions depending on the breeding/non-breeding season.  
It’d be interesting to look at in the occurrence data, since auk  
allows zero-filling.

Let’s have a look at threats data.

threats <- purrr::map\_df(species\_info, "threats")

str(threats)

## 'data.frame': 414 obs. of 8 variables:

## $ id : int 22696993 22696993 22696993 22696993 22696993 22696993 22696993 22696993 103888106 103888106 ...

## $ threat1 : chr "Agriculture & aquaculture" "Biological resource use" "Biological resource use" "Biological resource use" ...

## $ threat2 : chr "Annual & perennial non-timber crops" "Hunting & trapping terrestrial animals" "Hunting & trapping terrestrial animals" "Logging & wood harvesting" ...

## $ stresses: chr "Ecosystem degradation, Ecosystem conversion" "Species mortality" "Species mortality" "Species disturbance, Ecosystem degradation" ...

## $ timing : chr "Agriculture & aquaculture" "Biological resource use" "Biological resource use" "Biological resource use" ...

## $ scope : chr "Annual & perennial non-timber crops" "Hunting & trapping terrestrial animals" "Hunting & trapping terrestrial animals" "Logging & wood harvesting" ...

## $ severity: chr "Ongoing" "Ongoing" "Past, Likely to Return" "Ongoing" ...

## $ impact : chr "Ongoing" "Ongoing" "Past, Likely to Return" "Ongoing" ...

As you can see, it is a depressing dataset since most threats are  
human-made, but such information can help conservation! Instead of  
diving into the fate of particular species, let’s get a general picture.  
There are 211 species, for 68 we get an entry in the threats data. What  
are the most common threats for them, now and in the future?

dplyr::filter(threats, severity %in% c("Future", "Ongoing")) %>%

dplyr::select(id, threat2, threat1) %>%

dplyr::rename(threat\_category = threat1) %>%

dplyr::rename(threat\_subcategory = threat2) %>%

unique() %>%

dplyr::count(threat\_category, threat\_subcategory) %>%

dplyr::arrange(-n) %>%

head(n = 5) %>%

knitr::kable()

| **threat\_category** | **threat\_subcategory** | **n** |
| --- | --- | --- |
| Biological resource use | Hunting & trapping terrestrial animals | 25 |
| Climate change & severe weather | Habitat shifting & alteration | 24 |
| Energy production & mining | Renewable energy | 16 |
| Natural system modifications | Dams & water management/use | 14 |
| Pollution | Agricultural & forestry effluents | 14 |

It’s quite similar to the ranking in this post on [BirdLife’s  
website](https://www.birdlife.org/worldwide/news/top-five-threats-birds-may-surprise-you):  
industrial farming, logging, invasive species, hunting and trapping,  
climate change. Note that these threats have been summarized by species,  
not by species and location, so this data doesn’t tell us much about the  
state of each species in the County of Constance. If we wanted to  
characterize the County a bit more via the use of open data,

**Conclusion**

**Characterizing the local birds population**

From occurrence data we got names of species observed at least once in  
the area over the last year. Using taxonomy data to classify them we  
were able to see quite a few birds orders are represented, and we were  
able to visualize corresponding silhouettes. Traits data is general and  
doesn’t give information about the local context of birds in the County  
of Constance, however it could be coupled with more local data:  
localization of observations, and open geographical data. All the  
mentioned data sources are available for free, and can be obtained using  
R packages