Albacore Diet Synthesis A - Phylo Graphs

Natasha Hardy & Zachary Roote

15/04/2021

## Set-up your working environment

**Note:** Code issue has arisen discussed here: <https://stackoverflow.com/questions/26619434/install-an-old-version-of-dplyr-0-12-in-r> Here we will be updating some of the phylogenetic packages.

#Install packages if needed  
  
#remotes::install\_github("eliocamp/ggnewscale@dev")  
  
#remotes::install\_github("YuLab-SMU/ggtree")  
  
#if (!requireNamespace("BiocManager", quietly = TRUE))  
# install.packages("BiocManager")  
  
#BiocManager::install("Biostrings")  
#BiocManager::install("ggtree")

# general  
library(tidyverse)  
library(readxl)  
library(plyr)  
library(dplyr)  
library(devtools)  
library(here)  
"%notin%" = Negate('%in%')  
here::here()  
  
# graphics  
  
library(ggplot2)  
library(lattice)  
library(graphics)  
library(ggnewscale)  
library(ggimage)  
library(reshape2)  
library(pander)  
library(gridExtra)  
library(captioner)  
library(unmarked)  
library(cowplot)  
library(gtable)  
library(viridis)  
library(PNWColors)  
library(RColorBrewer)  
  
# phylogenetic & other tools  
library(ape)  
library(Biostrings)  
library(raster)  
library(dismo)  
library(patchwork)  
library(ggtree)  
library(ggstance)  
library(geiger)  
library(rotl) #for phylogenetic analyses, get all the species? from Hinchliff et al. 2015 PNAS  
library(phylobase)  
library(phytools)  
library(phangorn)  
library(stringr)  
library(taxize)  
library(treeio)

## Load Data

Using combination of my trial code and Matt Savoca and company’s code to build a phylo tree with associated trait and %FO data

#data contains sp list, class, order & family + traits.  
  
#Probable life stage data  
my\_prey\_prob = read.csv(here("./data/output\_data/prey\_probable\_traits.csv"), header=TRUE) %>%  
 dplyr::select(prey\_sp, prey\_class:prey\_family, life\_stage, vert\_habitat:maxM, -X) %>%  
 filter(prey\_sp %notin% c("Lampanyctus mexicanus", "Janthina exigua"))  
#298 species and 32 variables  
str(my\_prey\_prob)

## 'data.frame': 298 obs. of 32 variables:  
## $ prey\_sp : chr "Abralia redfieldi" "Abraliopsis affinis" "Abraliopsis felis" "Abraliopsis gilchristi" ...  
## $ prey\_class : chr "Cephalopoda" "Cephalopoda" "Cephalopoda" "Cephalopoda" ...  
## $ prey\_order : chr "Oegopsida" "Oegopsida" "Oegopsida" "Oegopsida" ...  
## $ prey\_family : chr "Enoploteuthidae" "Enoploteuthidae" "Enoploteuthidae" "Enoploteuthidae" ...  
## $ life\_stage : chr "adult" "adult" "adult" "adult" ...  
## $ vert\_habitat : chr "mesopelagic" "mesopelagic" "mesopelagic" "mesopelagic" ...  
## $ horz\_habitat : chr "oceanic" "oceanic" "oceanic" "continental slope" ...  
## $ diel\_migrant : int 1 1 1 1 1 1 1 NA 1 NA ...  
## $ diel\_migrant\_cat : chr "diel\_yes" "diel\_yes" "diel\_yes" "diel\_yes" ...  
## $ refuge : int 0 0 0 0 0 0 0 0 1 0 ...  
## $ refuge\_cat : chr "refuge\_no" "refuge\_no" "refuge\_no" "refuge\_no" ...  
## $ season\_migrant : int NA NA NA NA NA 1 NA 1 0 NA ...  
## $ season\_cat : chr "season\_NA" "season\_NA" "season\_NA" "season\_NA" ...  
## $ body\_shape : chr "fusiform" "fusiform" "fusiform" "fusiform" ...  
## $ l\_max : num 3.6 4.3 8 5.1 3.5 27 NA 7.8 0.43 8.2 ...  
## $ phys\_defense : int 0 0 0 0 0 1 0 0 1 0 ...  
## $ transparent : int 1 1 1 0 0 0 1 0 0 1 ...  
## $ col\_disrupt : int 1 1 1 0 0 0 0 1 0 0 ...  
## $ silver : int 0 0 0 0 0 0 1 0 0 0 ...  
## $ countershade : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ gregarious\_primary: chr NA NA NA NA ...  
## $ trophic\_level : num NA NA NA NA NA NA NA NA NA NA ...  
## $ fisheries\_status : chr "none" "none" "none" "none" ...  
## $ b\_shape\_r : num 3.91 5.35 5.28 5.63 8.39 ...  
## $ eye\_body\_r : num 0.0849 0.0698 0.0508 0.0501 0.0421 ...  
## $ standard\_total : num 0.527 0.457 0.458 0.422 0.349 ...  
## $ energy\_density : num NA NA 4.4 NA NA NA 1.7 3.9 NA NA ...  
## $ percent\_protein : num NA NA 17.4 NA NA NA NA 16.4 NA NA ...  
## $ percent\_lipid : num NA NA NA NA NA ...  
## $ maxFO : num 5 1.16 36.4 0.7 4.9 ...  
## $ maxN : num 0 0.0765 11.3 0.2 3.4 ...  
## $ maxM : num 0 0.00609 2.13235 0.1 3.4 ...

sapply(my\_prey\_prob, class)

## prey\_sp prey\_class prey\_order prey\_family   
## "character" "character" "character" "character"   
## life\_stage vert\_habitat horz\_habitat diel\_migrant   
## "character" "character" "character" "integer"   
## diel\_migrant\_cat refuge refuge\_cat season\_migrant   
## "character" "integer" "character" "integer"   
## season\_cat body\_shape l\_max phys\_defense   
## "character" "character" "numeric" "integer"   
## transparent col\_disrupt silver countershade   
## "integer" "integer" "integer" "integer"   
## gregarious\_primary trophic\_level fisheries\_status b\_shape\_r   
## "character" "numeric" "character" "numeric"   
## eye\_body\_r standard\_total energy\_density percent\_protein   
## "numeric" "numeric" "numeric" "numeric"   
## percent\_lipid maxFO maxN maxM   
## "numeric" "numeric" "numeric" "numeric"

## these need to be factors to work on the trees properly  
my\_prey\_prob[,c("diel\_migrant","refuge","season\_migrant", "phys\_defense", "transparent", "col\_disrupt", "silver", "countershade")] <- lapply(my\_prey\_prob[,c("diel\_migrant","refuge","season\_migrant", "phys\_defense", "transparent", "col\_disrupt", "silver", "countershade")], factor)  
  
my\_prey\_prob$maxFO[my\_prey\_prob$maxFO == 0.00] <- NA  
my\_prey\_prob$maxN[my\_prey\_prob$maxN == 0.00] <- NA  
my\_prey\_prob$maxM[my\_prey\_prob$maxM == 0.00] <- NA

**NOTES:** A lot of troubleshooting later and I found that the species list needs to go on the far left side of the data, all other data/factors need to be added to the RHS of the species list

## Tree Build

Building the basic tree

# Tree build 1  
breaks <- c(seq(1,nrow(my\_prey\_prob),50),nrow(my\_prey\_prob)+1) # why are we doing this?  
#looking up each of the species in dataset to a phylogeny, doing it by sets of 50  
#if there's an NA (species that didn't match a value in known phylogeny), breaks the whole chunk of 50 species  
  
for (i in 1:(length(breaks)-1)){  
 taxa <- as.character(my\_prey\_prob$prey\_sp[breaks[i]:(breaks[i+1]-1)])  
 taxa <- taxa[taxa != "" & !is.na(taxa)]  
  
 resolved\_namest <- tnrs\_match\_names(taxa) # I think this is where all the extra species fall out  
 resolved\_namest <- resolved\_namest[!is.na(resolved\_namest$unique\_name),] # ignore an NA  
 if (i==1){  
 resolved\_namess <- resolved\_namest  
 } else {  
 resolved\_namess <- rbind(resolved\_namess, resolved\_namest)  
 }  
}  
resolved\_names <- resolved\_namess  
resolved\_names <- resolved\_names[resolved\_names$flags!="INCERTAE\_SEDIS\_INHERITED",]  
  
#original tree based on simple taxon datset  
my\_tree\_prob <- tol\_induced\_subtree(ott\_ids = resolved\_names$ott\_id, label\_format = "name")  
  
my\_tree\_prob$tip.label<-gsub("\_"," ",my\_tree\_prob$tip.label) # removes underscore between genus and species names  
my\_tree\_prob$tip.label<-str\_extract(my\_tree\_prob$tip.label, "[A-Z][a-z]+ [a-z]+")  
  
my\_tree\_prob <- compute.brlen(my\_tree\_prob, method = "Grafen", power = 1/2) #add branch lengths to my tree using the Grafen (1989) method  
my\_tree\_prob <- ladderize(my\_tree\_prob, right = TRUE)  
  
View(my\_tree\_prob)  
  
# SAVE TREE ----  
write.tree(my\_tree\_prob, here("./data/output\_data/albacore\_diet\_tree\_prob"))

my\_tree\_prob <- read.tree(here("./data/output\_data/albacore\_diet\_tree\_prob"))

## Sort out the tree for graphing

#Edit tip labels  
my\_tree\_prob$tip.label <- as.factor(sub("\_", " ", my\_tree\_prob$tip.label))  
  
nrow(my\_prey\_prob) == length(my\_tree\_prob$tip.label) #TRUE

## [1] TRUE

nrow(my\_prey\_prob) #298

## [1] 298

length(my\_tree\_prob$tip.label) #298

## [1] 298

**NOTE:** We have in the past encountered a data frame dimension issue when plotting our data values onto the tree. There are also several species names that do not line up. This issue is addressed in the chunk below.

#my\_prey\_prob$prey\_sp <- gsub(" ", "\_", my\_prey\_prob$prey\_sp) #for ease of plotting take away " "  
  
tree\_names\_prob = data.frame(sort(my\_tree\_prob$tip.label)) #sort the species names from the phylo tree  
#Still 301 spp  
## sort brings tree\_names\_prob from 301 to 300 because of an NA in the tip labels  
prey\_names\_prob = data.frame(sort(my\_prey\_prob$prey\_sp)) #sort the species names from the original list  
  
nrow(tree\_names\_prob); nrow(prey\_names\_prob) #one species didn't make it to the tree

## [1] 298

## [1] 298

## here we check the names that are in the prey data but not on the tip labels of the tree  
names\_not\_in\_tree\_prob = prey\_names\_prob %>%  
 filter(sort.my\_prey\_prob.prey\_sp. %notin% tree\_names\_prob$sort.my\_tree\_prob.tip.label.)  
  
## check names that are in the tree tip labels but not in the prey data  
names\_not\_in\_prey\_prob = tree\_names\_prob %>%  
 filter(sort.my\_tree\_prob.tip.label. %notin% prey\_names\_prob$sort.my\_prey\_prob.prey\_sp.)  
  
## here we decide to match the names in the prey data to the tip labels, so we create a dataset with the names in the prey data that are in the tree (I see this looks kind of confusing with the double negative, the names not not in the tree). These are the names that do not need to be fixed, which we will bind the fixed names to afterwards so we do not get doubles of the same names (with different spelling).  
my\_prey\_keep\_prob = my\_prey\_prob %>%  
 filter(prey\_sp %notin% names\_not\_in\_tree\_prob$sort.my\_prey\_prob.prey\_sp.)  
  
## these are the names we are going to fix to match the tip labels of the tree, given they are synonyms/misspellings. These are the same names in 'names\_not\_in\_tree\_adult', except they now have the columns of data which will allow use to merge it to 'my\_prey\_keep\_adult' after we fix the names.  
my\_prey\_fix\_prob = my\_prey\_prob %>%  
 filter(prey\_sp %in% names\_not\_in\_tree\_prob$sort.my\_prey\_prob.prey\_sp.)  
## this step isn't necessary, but it does make it easier in the names are in alphabetical order  
my\_prey\_fix\_prob = my\_prey\_fix\_prob[order(my\_prey\_fix\_prob$prey\_sp),]  
# the names have to be characters for the name reassignment to work, it will give an error if we don't do this (they are factors before this)  
my\_prey\_fix\_prob$prey\_sp <- as.character(my\_prey\_fix\_prob$prey\_sp)  
## we make these into x2 and y2 because it makes it much easier to write the next section  
x2 = my\_prey\_fix\_prob  
y2 = as.vector(names\_not\_in\_prey\_prob$sort.my\_tree\_prob.tip.label.)  
  
#here we look at the names in both of the lists and see if there are misspellings/synonyms in the names that caused the discrepancy between the name lists. As you can see this is true for all of the species in these lists expect one. 'Diplodus sargus' does not have a match in the tree, since the only other tip labels left in the tree is an NA after the rest of the reassignments.  
# if you want to see how this works you can run individual pieces of this before running the whole thing. For example x2[4,1] is "Leuroglossus stilbius", and y2[1] is "Bathylagus stilbius". Then after running this code we match the name in the prey data (x2) to the names in the tree (y2), so "Leuroglossus stilbius" becomes "Bathylagus stilbius".  
x2[1,1] = y2[4]; x2[2,1] = y2[1]; x2[3,1] = y2[2]; x2[4,1] = y2[3];   
x2[5,1] = y2[5]; x2[6,1] = y2[7]; x2[7,1] = y2[6];  
  
# we need to do the opposite for one of the names here "Diplodus sargus", which is in the prey data but not in the tip labels. So, we are replacing the NA in the tip labels  
tip\_labels\_prob <- as.vector(my\_tree\_prob$tip.label)  
tip\_labels\_prob[28] ##this is the NA we want to replace

## [1] "NA"

tip\_labels\_prob[28] = "Diplodus sargus"  
## replacing the tip labels with the additional name  
my\_tree\_prob$tip.label <- tip\_labels\_prob  
  
# here we bind the fixed names to the names that didnt need to be fixed  
my\_prey\_prob = rbind(my\_prey\_keep\_prob, x2)  
#here we get rid of any of the names in the data that didn't have a match in the tree tip labels, which was only 'Diplodus sargus'  
my\_prey\_prob = my\_prey\_prob %>%  
 filter(prey\_sp %in% my\_tree\_prob$tip.label)  
nrow(my\_prey\_prob);length(my\_tree\_prob$tip.label) ## one of those tip labels is an NA

## [1] 297

## [1] 298

#the prey names need to be the rownames for the heatmap to work on the tree  
rownames(my\_prey\_prob) <- my\_prey\_prob$prey\_sp  
  
#tip labels need to be characters to pipe onto the tree  
my\_tree\_prob$tip.label <- as.character(my\_tree\_prob$tip.label)  
str(my\_tree\_prob)

## List of 5  
## $ edge : int [1:562, 1:2] 299 300 301 302 303 304 305 306 307 308 ...  
## $ edge.length: num [1:562] 0.00168 0.24846 0.00225 0.01824 0.03549 ...  
## $ Nnode : int 265  
## $ node.label : chr [1:265] "mrcaott42ott150" "mrcaott42ott49" "mrcaott42ott658" "Clupeocephala" ...  
## $ tip.label : chr [1:298] "Sebastes wilsoni" "Sebastes zacentrus" "Sebastes proriger" "Sebastes brevispinis" ...  
## - attr(\*, "class")= chr "phylo"  
## - attr(\*, "order")= chr "cladewise"

## Paper figures - the list

### Basic phylogenies

**Basic prey species phylogeny graph**

#Phylotree without labels  
prob\_basic = ggtree(my\_tree\_prob, layout = 'circular')  
  
#Phylotree with species labels  
prob\_basic\_lab = ggtree(my\_tree\_prob, layout = 'circular') + geom\_tiplab(size = 2)

**Class-based phylogeny graph**

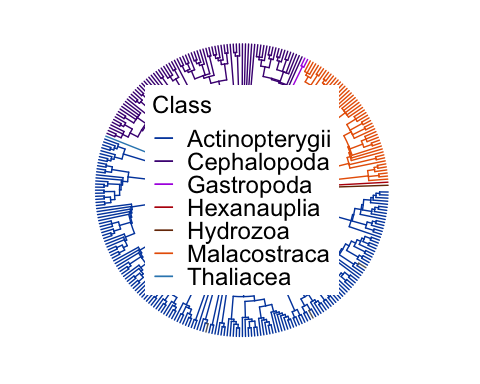
#Use basic + prey\_sp labels + prey\_class  
  
# creating a dataset that splits the species data by class  
prey\_class\_prob\_info <- split(x = my\_prey\_prob$prey\_sp, f = my\_prey\_prob$prey\_class)  
unique(my\_prey\_prob$prey\_class)

## [1] "Cephalopoda" "Malacostraca" "Actinopterygii" "Gastropoda"   
## [5] "Hexanauplia" "Hydrozoa" NA "Thaliacea"

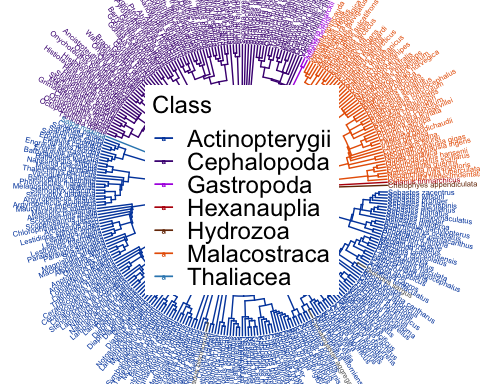
# splitting the tree species by class  
my\_tree\_prob\_class <- groupOTU(my\_tree\_prob, prey\_class\_prob\_info)  
as.character(sort(unique(my\_prey\_prob$prey\_class)))

## [1] "Actinopterygii" "Cephalopoda" "Gastropoda" "Hexanauplia"   
## [5] "Hydrozoa" "Malacostraca" "Thaliacea"

#creating a palette for the class split  
#pal\_prob\_class = c("grey40", '#0047ab', '#751308', '#4B0082', '#F05E23', '#013220', '#B80F0A', "#66C2A5", "#3288BD") ## the grey40 is needed for a branch between branches, but doesn't assign to a class  
  
  
pal\_prob\_class = c("grey40", ## the grey40 is needed for a branch between branches, but doesn't assign to a class  
 '#0047ab', #Actinopterygii  
 #'#751308', #Branchiopoda #older teal colour "#66C2A5" #Not in prob data just was used for adult data  
 '#4B0082', #Cephalopoda  
 "#AD15E2", #Gastropoda #old orange '#F05E23'  
 '#B80F0A', #Hexanauplia  
 "#773405", #Hydrozoa #'#013220'  
 "#E86103", #Malacostraca '  
 "#3288BD" #Thaliacea  
 )  
  
## plotting the tree without tip labels  
  
prob\_class <- ggtree(my\_tree\_prob\_class, aes(color = group), layout = 'circular') +  
 theme(legend.position = c(0.5,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_colour\_manual('Class', aesthetics = c('colour'), values = pal\_prob\_class,  
 breaks = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa", "Malacostraca","Thaliacea"),  
 labels = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa", "Malacostraca","Thaliacea"))  
  
prob\_class

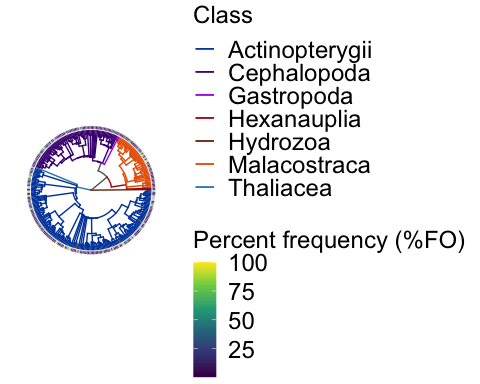


# plotting the tree with tip labels  
prob\_class\_lab <- ggtree(my\_tree\_prob\_class, aes(color = group), layout = 'circular') +  
 theme(legend.position = c(0.5,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_colour\_manual('Class', aesthetics = c('colour', 'fill'), values = pal\_prob\_class,  
 breaks = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa", "Malacostraca","Thaliacea"),  
 labels = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa", "Malacostraca","Thaliacea")) +  
 geom\_tiplab(size = 2)  
  
prob\_class\_lab

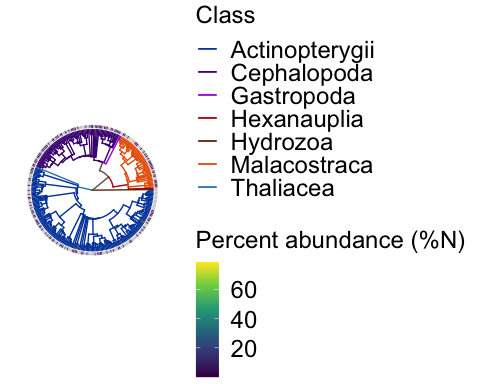


### Diet use data

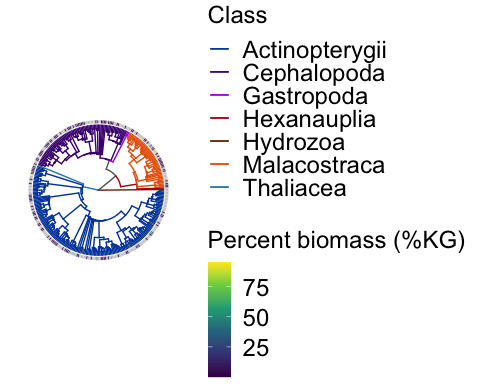
#Overlay maxFO (frequency of occurrence data) on a basic prey tree coloured by Class  
#Use prey\_class, maxFO  
  
prob\_maxfo <- gheatmap(prob\_class, my\_prey\_prob[,"maxFO", drop = FALSE],   
 offset = 0, width = 0.05, colnames = FALSE) +  
 theme(#legend.position = c(0.5,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_viridis\_c(name = "Percent frequency (%FO)", na.value = 'grey')  
  
prob\_maxfo



#Overlay maxN (abundance data) on a basic prey tree coloured by Class  
#Use prey\_class, maxN  
  
prob\_maxn <- gheatmap(prob\_class, my\_prey\_prob[,"maxN", drop = FALSE],   
 offset = 0, width = 0.05, colnames = FALSE) +  
 theme(#legend.position = c(0.5,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_viridis\_c(name = "Percent abundance (%N)", na.value = 'grey')  
prob\_maxn



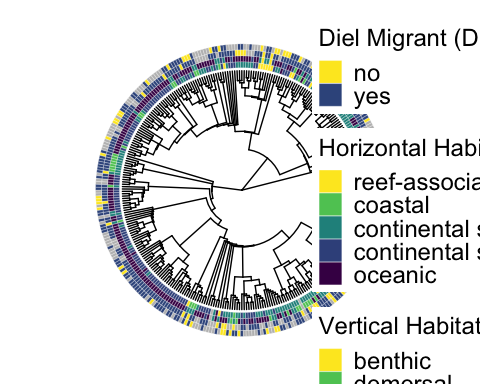
#Overlay maxM (biomass data) on a basic prey tree coloured by Class  
#Use prey\_class, maxM  
  
prob\_maxm <- gheatmap(prob\_class, my\_prey\_prob[,"maxM", drop = FALSE],   
 offset = 0, width = 0.05, colnames = FALSE) +  
 theme(#legend.position = c(0.5,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_viridis\_c(name = "Percent biomass (%KG)", na.value = 'grey')  
prob\_maxm



### Trait data

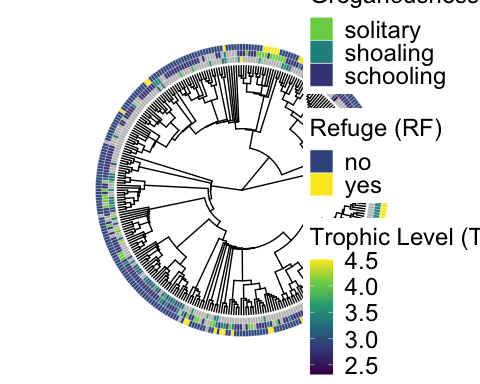
For habitat use traits: we are going to plot multiple rings of data around the phylogenetic trees using vert\_habitat, horz\_habitat, diel\_migrant (change to yes/no), and season\_migrant (yes/no)

#creating this plot one iteration at a time. Adding one layer of the heatmap, then allowing for another legend on the plot, and then another layer of the heatmap  
  
habitat\_traits\_prob1 <- gheatmap(prob\_basic, my\_prey\_prob[, 'vert\_habitat',drop=FALSE],   
 offset= 0, width=0.05, colnames = FALSE) +  
 scale\_fill\_viridis\_d(name = "Vertical Habitat (VH)",   
 direction = -1,   
 breaks = c("benthic", "demersal", "epipelagic", "mesopelagic", "bathypelagic"),  
 limits = c("benthic", "demersal", "epipelagic", "mesopelagic", "bathypelagic"),   
 na.translate = TRUE,   
 option = 'D',   
 na.value = 'grey')+  
 theme(legend.position = c(1,0.80),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,4))+ #this opens up the gap for the Vertical habitat label around the tree  
 annotate('text', x = 1.03, y = -7, label = 'VH', angle = -85, size = 4)  
  
habitat\_traits\_prob1.5 <- habitat\_traits\_prob1 + new\_scale\_fill()  
  
habitat\_traits\_prob2 <- gheatmap(habitat\_traits\_prob1.5, my\_prey\_prob[ ,'horz\_habitat',drop=FALSE],   
 offset=0.05, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Horizontal Habitat (HH)",   
 option = "D",   
 direction = -1,  
 breaks = c("reef-associated", "coastal", "continental shelf","continental slope", "oceanic"),  
 limits = c("reef-associated", "coastal", "continental shelf","continental slope", "oceanic"),  
 na.translate = TRUE,   
 na.value = 'grey')+  
 theme(legend.position = c(1,0.683),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,4))+  
annotate('text', x = 1.08, y = -7.1, label = 'HH', angle = -85, size = 4)  
  
habitat\_traits\_prob2.5 <- habitat\_traits\_prob2 + new\_scale\_fill()  
  
habitat\_traits\_prob3 <- gheatmap(habitat\_traits\_prob2.5, my\_prey\_prob[ , 'diel\_migrant',drop=FALSE], offset=0.10, width=0.05, colnames = F) +  
 scale\_fill\_manual(name = "Diel Migrant (DM)",  
 breaks = c("0", "1"),  
 limits = c("0", "1"),  
 labels = c("no","yes"),  
 values = c("0"="#FDE725FF", "1"="#39568CFF"),   
 na.value = 'grey')+  
 theme(legend.position = c(1,0.5985),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,4))+  
 annotate('text', x = 1.13, y = -7.5, label = 'DM', angle = -85, size = 4)  
  
habitat\_traits\_prob3.5 <- habitat\_traits\_prob3 + new\_scale\_fill()  
  
habitat\_traits\_prob\_final <- gheatmap(habitat\_traits\_prob3.5,   
 my\_prey\_prob[ ,'season\_migrant',drop=FALSE],  
 offset=0.15, width=0.05, colnames = F) +  
 theme(legend.position = c(1.05,0.50),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_manual(name = "Seasonal Migrant (SM)",  
 breaks = c("0", "1"),  
 limits = c("0", "1"),  
 labels = c("no", "yes"),  
 values = c("0"="#FDE725FF", "1"="#39568CFF"),   
 na.value = 'grey') +  
 scale\_y\_continuous(expand = c(0,4))+  
 annotate('text', x = 1.18, y = -7.5, label = 'SM', angle = -85, size = 4)  
  
   
#Check graph  
habitat\_traits\_prob\_final



For seasonal, trophic & aggregation behaviour use: trophic\_level, gregarious\_primary and refuge\_use (yes/no)

behav\_trophic\_prob1 <- gheatmap(prob\_basic, my\_prey\_prob[,'trophic\_level', drop=FALSE],   
 offset=0, width=0.05, colnames = F) +  
 theme(legend.position = c(1.05,0.80),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_viridis\_c(name = "Trophic Level (TL)", na.value = 'grey') +  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.04, y = -6.5, label = 'TL', angle = -85, size = 4)  
  
behav\_trophic\_prob1.5 <- behav\_trophic\_prob1 + new\_scale\_fill()  
  
behav\_trophic\_prob2 <- gheatmap(behav\_trophic\_prob1.5, my\_prey\_prob[,"gregarious\_primary",drop=FALSE],   
 offset=0.05, width=0.05, colnames = F) +  
 theme(legend.position = c(1.05,0.65),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_fill\_viridis\_d(name = "Gregariousness (GG)",   
 direction = -1,  
 breaks = c("solitary", "shoaling","schooling"),  
 limits = c("solitary", "shoaling","schooling"),  
 na.translate = TRUE,  
 option = "D", #'magma',  
 begin = 0.2,end = 0.8,  
 guide = guide\_legend(order = 1),   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.09, y = -6.5, label = 'GG', angle = -85, size = 4)  
  
behav\_trophic\_prob2.5 <- behav\_trophic\_prob2 + new\_scale\_fill()  
  
behav\_trophic\_prob\_final <- gheatmap(behav\_trophic\_prob2.5,   
 my\_prey\_prob[ ,'refuge',drop=FALSE],   
 offset=0.10, width=0.05, colnames = F)+   
 scale\_fill\_manual(name = "Refuge (RF)",  
 breaks = c("0", "1"),  
 limits = c("0", "1"),  
 labels = c("no", "yes"),  
 values = c("1"="#FDE725FF", "0"="#39568CFF"),  
 na.value = 'grey')+  
 theme(legend.position = c(1,0.531),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,4))+  
 annotate('text', x = 1.14, y = -6.5, label = 'RF', angle = -85, size = 4)  
  
behav\_trophic\_prob\_final



For morphological (shape) traits use: body\_shape, b\_shape\_r and eye\_body\_r

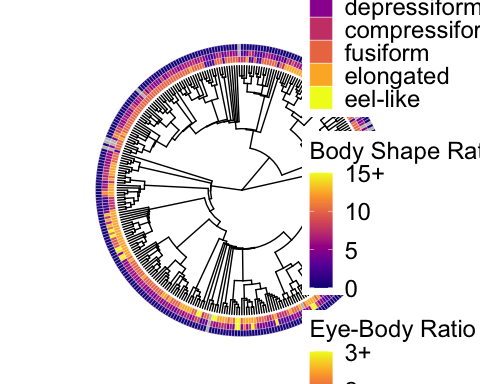
levels(as.factor(my\_prey\_prob$body\_shape))

## [1] "compressiform" "depressiform" "eel-like" "elongated"   
## [5] "fusiform" "globiform" "unique"

my\_prey\_prob$body\_shape <- as.factor(my\_prey\_prob$body\_shape)  
my\_prey\_prob$body\_shape <- factor(my\_prey\_prob$body\_shape,  
 levels = c("unique", "globiform", "depressiform", "compressiform",  
 "fusiform", "elongated", "eel-like"))  
levels(my\_prey\_prob$body\_shape)

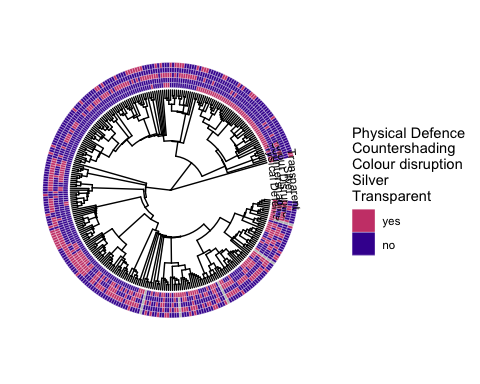
## [1] "unique" "globiform" "depressiform" "compressiform"  
## [5] "fusiform" "elongated" "eel-like"

morphology\_comparison\_prob1 <- gheatmap(prob\_basic,   
 my\_prey\_prob[ ,'body\_shape',drop=FALSE],  
 offset=0, width=0.05,font.size=2, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Body Shape (BD)",   
 option = 'C',  
 breaks = c("unique", "globiform", "depressiform", "compressiform",  
 "fusiform", "elongated", "eel-like"),  
 limits = c("unique", "globiform", "depressiform", "compressiform",  
 "fusiform", "elongated", "eel-like"),  
 guide = guide\_legend(order = 1,   
 #reverse = TRUE  
 ), #order = 1  
 na.value = 'grey')+  
 theme(legend.position = c(1.05,0.71),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.04, y = -6.5, label = 'BD', angle = -85, size = 4)  
  
morphology\_comparison\_prob1.5 <- morphology\_comparison\_prob1 + new\_scale\_fill()  
  
morphology\_comparison\_prob2 <- gheatmap(morphology\_comparison\_prob1.5,   
 my\_prey\_prob[ ,'b\_shape\_r',drop=FALSE],   
 offset=0.05, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_c(name = "Body Shape Ratio (BDR)",   
 option = 'C', limits=c(0,15),   
 oob = scales::squish,   
 breaks = c(0,5,10,15),   
 labels = c(0,5,10,"15+"),   
 na.value = 'grey')+  
 theme(legend.position = c(1.05,0.605),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.095, y = -6.5, label = 'BDR', angle = -85, size = 4)  
  
morphology\_comparison\_prob2.5 <- morphology\_comparison\_prob2 + new\_scale\_fill()  
  
morphology\_comparison\_prob\_final <- gheatmap(morphology\_comparison\_prob2.5,   
 my\_prey\_prob[ ,'eye\_body\_r',drop=FALSE],   
 offset=0.10, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_c(name = "Eye-Body Ratio (EBR)",   
 option = 'C',  
 limits=c(0,3),   
 oob = scales::squish,   
 breaks = c(0,1,2,3),  
 labels = c(0,1,2,"3+"),   
 na.value = 'grey')+  
 theme(legend.position = c(1.05,0.5),  
 legend.title = element\_text(size = 18),   
 legend.text = element\_text(size = 18))+  
 scale\_y\_continuous(expand = c(0,3.5))+  
annotate('text', x = 1.15, y = -6.5, label = 'EBR', angle = -85, size = 4)  
  
morphology\_comparison\_prob\_final



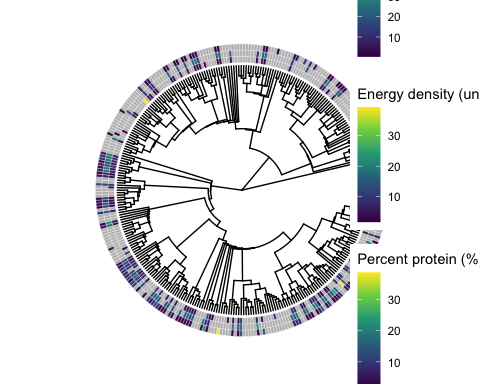
For morphological (defense/avoidance) traits use: phys\_defense, countershade, col\_disrupt, silver and transparent (in that order)

morphology\_binary\_prob1 <- gheatmap(prob\_basic, my\_prey\_prob[ ,'phys\_defense',drop=FALSE], offset=0, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Physical Defence", #"\nCountershading\nColour disruption\nSilver\nTransparent",  
 option = 'C',  
 breaks = c("1", "0"),  
 limits = c("1", "0"),  
 labels = c("yes", "no"),  
 begin=0.5, end=0.1,   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,4.5))+  
 annotate('text', x = 1.04, y = -8.5, label = 'Physical Defence', angle = -85, size = 3)  
  
morphology\_binary\_prob2 <- gheatmap(morphology\_binary\_prob1, my\_prey\_prob[ ,'countershade',drop=FALSE],   
 offset=0.05, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Physical Defence\nCountershading", #"\nColour disruption\nSilver\nTransparent",   
 option = 'C',  
 breaks = c("1", "0"),  
 limits = c("1", "0"),  
 begin=0.5, end=0.1,   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,4.5))+  
 annotate('text', x = 1.09, y = -8.5, label = 'Countershading', angle = -85, size = 3)  
  
morphology\_binary\_prob3 <- gheatmap(morphology\_binary\_prob2, my\_prey\_prob[ ,'col\_disrupt',drop=FALSE],  
 offset=0.10, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Physical Defence\nCountershading\nColour disruption", #"\nSilver\nTransparent",   
 option = 'C',  
 breaks = c("1", "0"),  
 limits = c("1", "0"),  
 labels = c("yes", "no"),  
 begin=0.5, end=0.1,   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,4.5))+  
 annotate('text', x = 1.14, y = -8.5, label = 'Colour Disruption', angle = -85, size = 3)  
  
morphology\_binary\_prob4 <- gheatmap(morphology\_binary\_prob3, my\_prey\_prob[ ,'silver',drop=FALSE],   
 offset=0.15, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Physical Defence\nCountershading\nColour disruption\nSilver", #"\nTransparent",   
 option = 'C',  
 breaks = c("1", "0"),  
 limits = c("1", "0"),  
 labels = c("yes", "no"),  
 begin=0.5, end=0.1,   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,4.5))+  
 annotate('text', x = 1.19, y = -8.5, label = 'Silver', angle = -85, size = 3)  
  
morphology\_binary\_prob\_final <- gheatmap(morphology\_binary\_prob4, my\_prey\_prob[ ,'transparent',drop=FALSE],   
 offset=0.20, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Physical Defence\nCountershading\nColour disruption\nSilver\nTransparent",   
 option = 'C',  
 breaks = c("1", "0"),  
 limits = c("1", "0"),  
 labels = c("yes", "no"),  
 begin=0.5, end=0.1,   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,4.5))+  
 annotate('text', x = 1.24, y = -8.5, label = 'Transparent', angle = -85, size = 3)  
  
morphology\_binary\_prob\_final



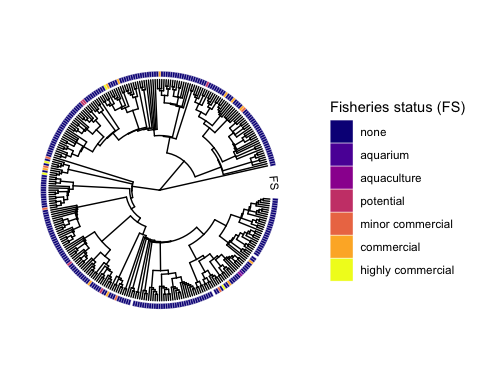
For Prey nutritional quality use: energy\_density, percent\_protein, percent\_lipid

nutritional\_quality\_prob1 <- gheatmap(prob\_basic, my\_prey\_prob[,'energy\_density',drop=FALSE],   
 offset=0, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_c(name = "Energy density (unit)",   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.04, y = -6.5, label = 'Energy density', angle = -85, size = 3)  
  
nutritional\_quality\_prob1.5 <- nutritional\_quality\_prob1 + new\_scale\_fill()  
  
nutritional\_quality\_prob2 <- gheatmap(nutritional\_quality\_prob1.5, my\_prey\_prob[,"percent\_protein",drop=FALSE],   
 offset=0.05, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_c(name = "Percent protein (%)",   
 na.value = 'grey')+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.09, y = -6.5, label = 'Percent protein (%)', angle = -85, size = 3)  
  
nutritional\_quality\_prob2.5 <- nutritional\_quality\_prob2 + new\_scale\_fill()  
  
nutritional\_quality\_prob\_final <- gheatmap(nutritional\_quality\_prob2.5, my\_prey\_prob[ ,'percent\_lipid',drop=FALSE],   
 offset=0.10, width=0.05, colnames = F) +  
 scale\_fill\_viridis\_c(name = "Percent lipid (%)",   
 na.value = 'grey') +  
 theme(legend.position = c(1,0.5985))+  
 scale\_y\_continuous(expand = c(0,4))+  
 annotate('text', x = 1.14, y = -6.5, label = 'Percent lipid (%)', angle = -85, size = 3)  
  
nutritional\_quality\_prob\_final



For Fisheries status use: fisheries\_status

fishery\_prob <- gheatmap(prob\_basic, my\_prey\_prob[ ,'fisheries\_status',drop=FALSE],   
 offset=0, width=0.05,font.size=2, colnames = F) +  
 scale\_fill\_viridis\_d(name = "Fisheries status (FS)",   
 option = 'C',  
 breaks = c("none","aquarium", "aquaculture", "potential","minor commercial",  
 "commercial", "highly commercial"),  
 limits = c("none","aquarium", "aquaculture", "potential","minor commercial",  
 "commercial", "highly commercial"))+  
 scale\_y\_continuous(expand = c(0,3.5))+  
 annotate('text', x = 1.04, y = -6.5, label = 'FS', angle = -85, size = 3)  
  
fishery\_prob



### EXTRAS

**Prey Order & Fam – TO DO:**

Code for these graphs are included in the .Rmd and not in the published document. These graphs are currently a lower priority and we may not get to this, however we include the code for anyone interested. The following would need to be fixed to improve the graphs:

1. Legends are too large as they are. I suggest not changing the colours as they clearly show different groups just not which ones are which using the legend. Is there any way to plot the “order” names near the branch of the tree or in a ring around the tree?
2. Also fix legend labels.

## SAVING OUTPUT

There are numerous issues with saving output either via ggsave() or dev.copy2pdf() functions. I’m batching the output saves below and doing all at once.

**Basic phylogeny**

Including code for saving this graph. Code for saving output for all other figures below are included in the .Rmd and not in the rendered document.

#Basic phylogenies  
  
#ggsave("prob\_basic.png",  
 #here('./output\_figures/phylos/prey\_prob/prob\_basic.png'),   
# plot=prob\_basic, width=8, height=8, dpi=300)  
  
#ggsave("prob\_basic\_lab.png",  
 #here('./output\_figures/phylos/prey\_prob/prob\_basic\_lab.png'),   
# plot=prob\_basic\_lab, width=10, height=10, dpi=300)

**Phylogeny by Class**

**Phylogeny by diet use**

**Phylogeny by habitat Use Traits**

**Phylogeny by behaviour & habitat use**

**Phylogeny by morphology - shape**

**Phylogeny by defensive morphology**

**Phylogeny by nutritional quality**

**Phylogeny by fisheries status**