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")

#For document rendering
#tinytex::install_tinytex()
```

```
# general
library(tidyverse)
library(readxl)
library(plyr)
library(dplyr)
library(devtools)
library(here)
"%notin%" = Negate('%in%')
here::here()
library(tinytex)

# 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" "Abraliops
##  $ 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 ...
```

```r
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"
```

```r
## 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",

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

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 ex
  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 spe
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 t
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
```

```r
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.

```r
#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
```

```r
## 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
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,
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
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
# if you want to see how this works you can run individual pieces of this before running the whole thin
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
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
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 bro
##  - 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

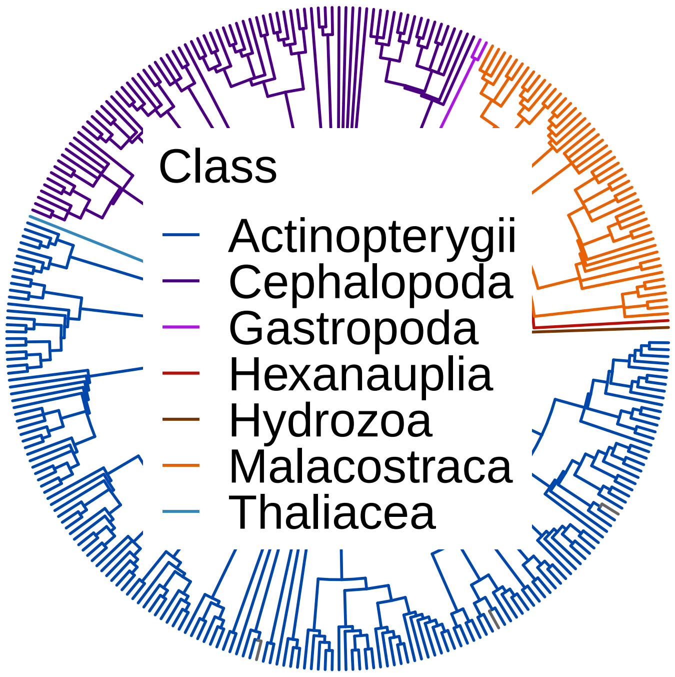
```
#Use basic + prey_sp labels + prey_class

# 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"
```

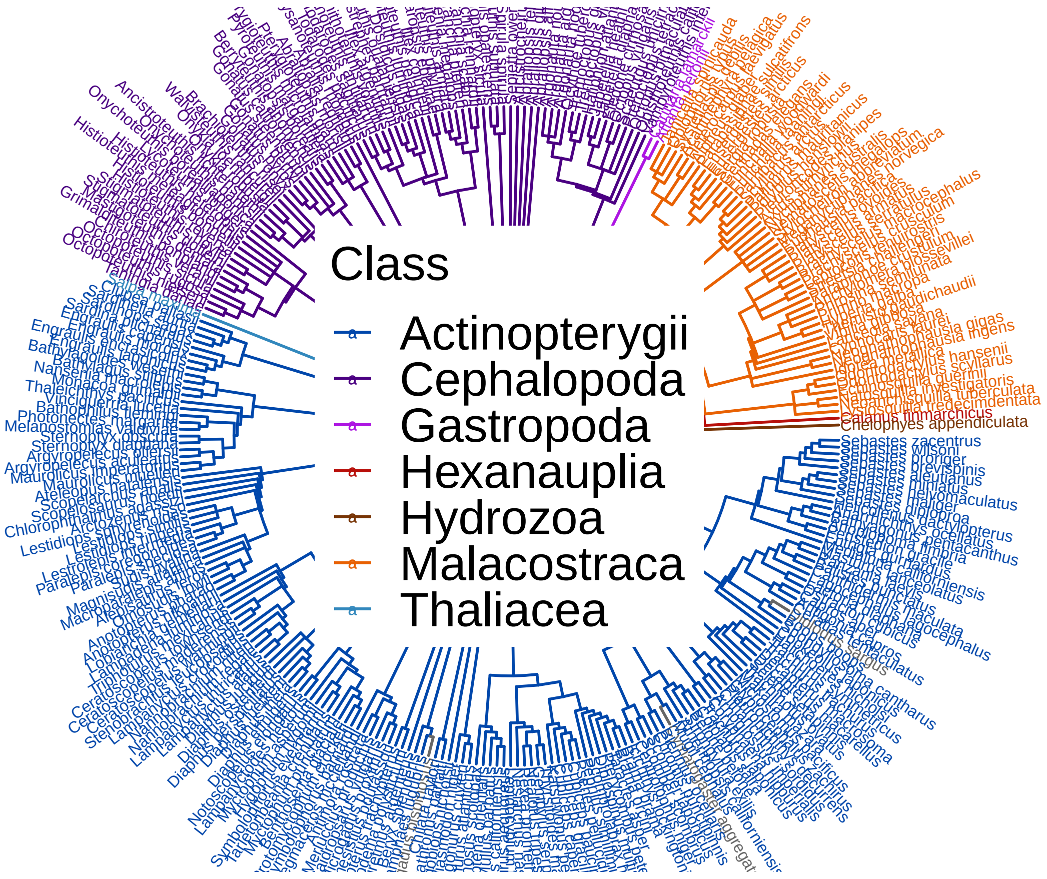
```
# splitting the tree species by class
my_tree_prob_class <- groupOTU(my_tree_prob, prey_class_prob_info)
as.character(sort(unique(my_prey_prob$prey_class)))
```

```
## [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', "#66C2A


pal_prob_class = c("grey40", ## the grey40 is needed for a branch between branches, but doesn't assign t
                   '#0047ab', #Actinopterygii
                   #'#751308', #Branchiopoda #older teal colour "#66C2A5" #Not in prob data just was us
                   '#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
                      labels = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa

prob_class
```

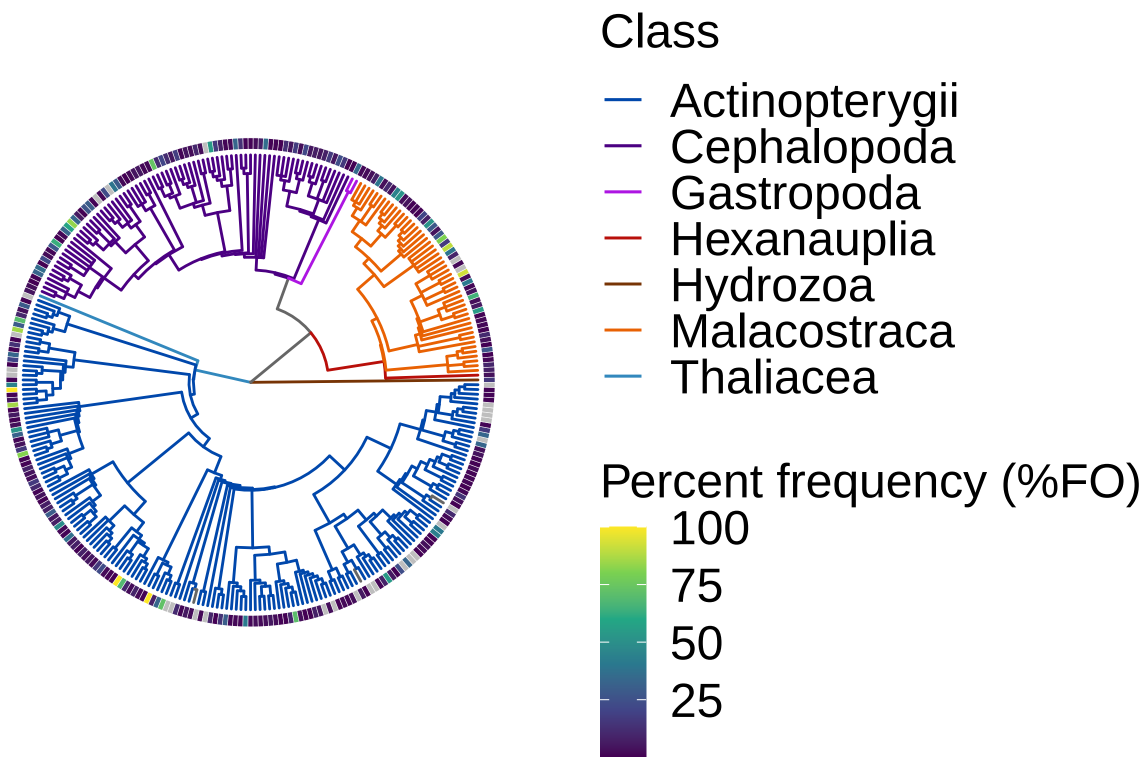


```
# plotting the tree with tip labels
prob_class_lab <- ggtree(my_tree_prob_class, aes(color = group), layout = 'circular') +
   theme(legend.position = c(0.5,0.50),
        legend.title = element_text(size = 18),
        legend.text = element_text(size = 18))+
  scale_colour_manual('Class', aesthetics = c('colour', 'fill'), values = pal_prob_class,
                      breaks = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa
                      labels = c("Actinopterygii", "Cephalopoda","Gastropoda", "Hexanauplia", "Hydrozoa
  geom_tiplab(size = 2)

prob_class_lab
```

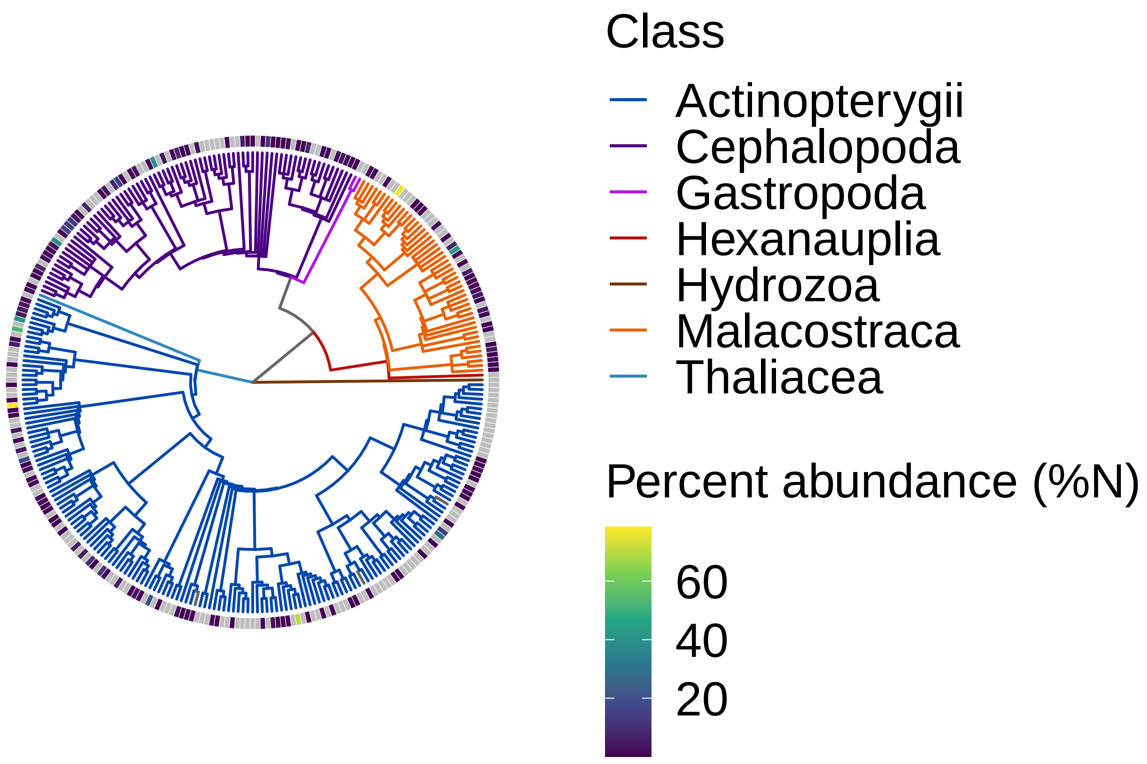


Diet use data

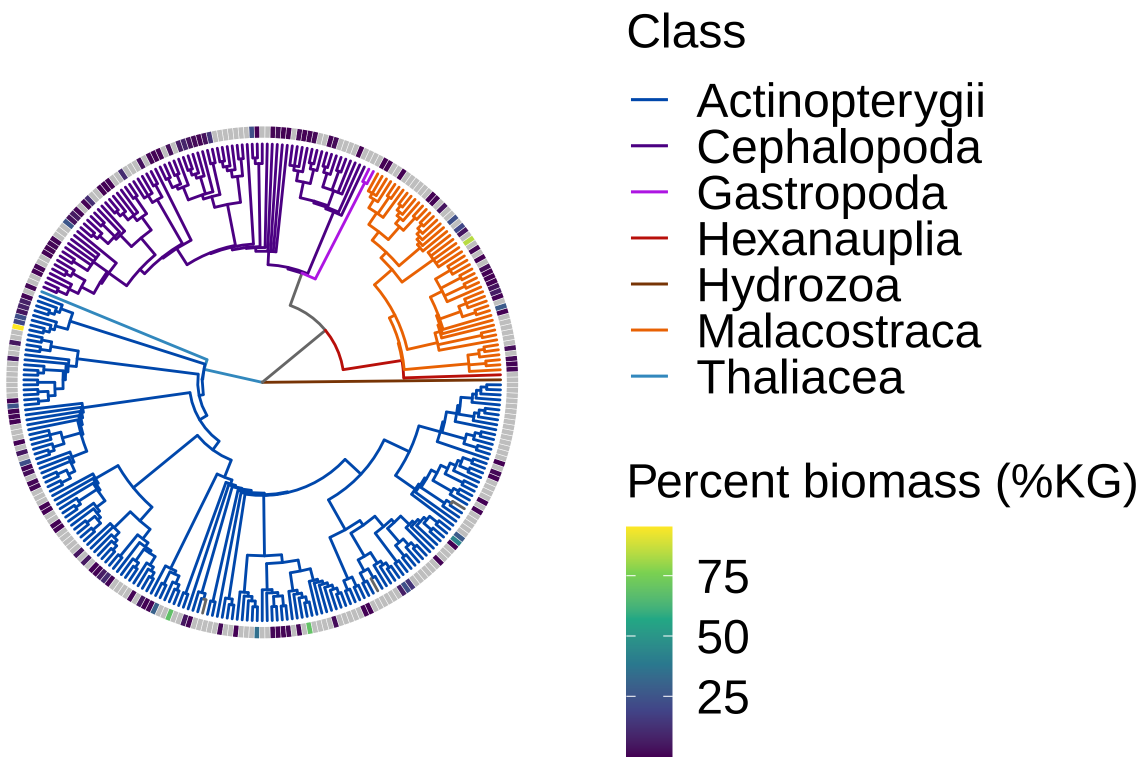
```
#Overlay maxFO (frequency of occurrence data) on a basic prey tree coloured by Class
#Use prey_class, maxFO

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
```



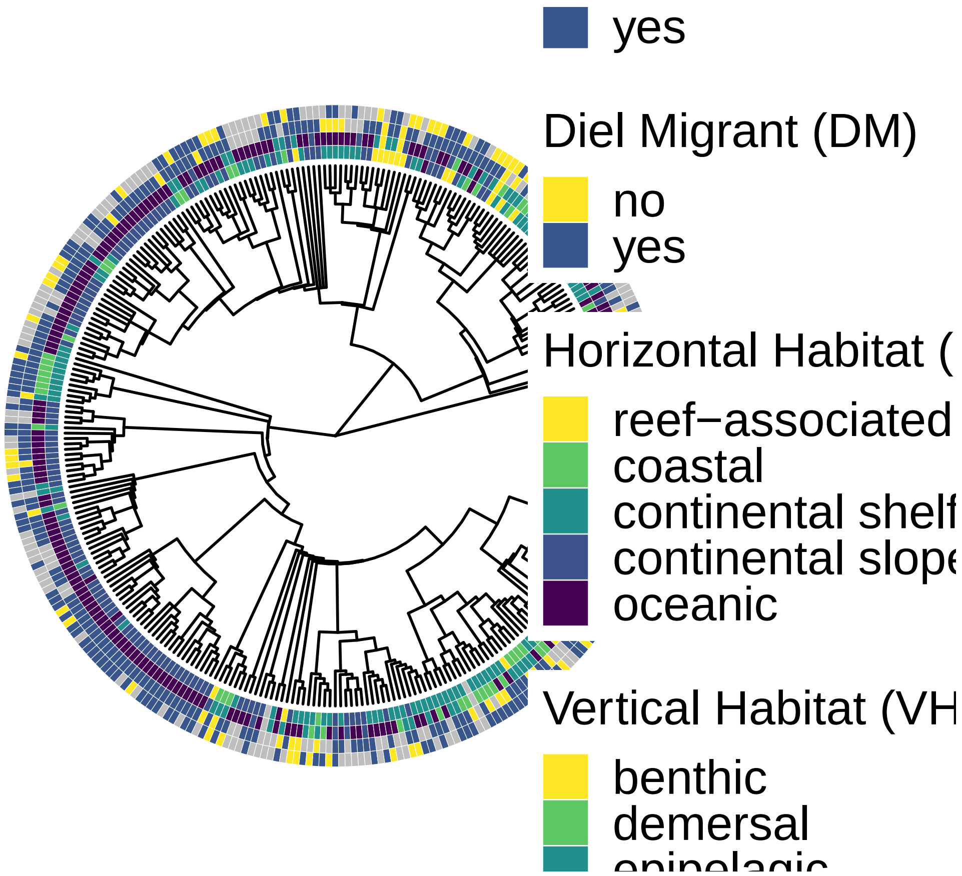
```
#Overlay maxM (biomass data) on a basic prey tree coloured by Class
#Use prey_class, maxM

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

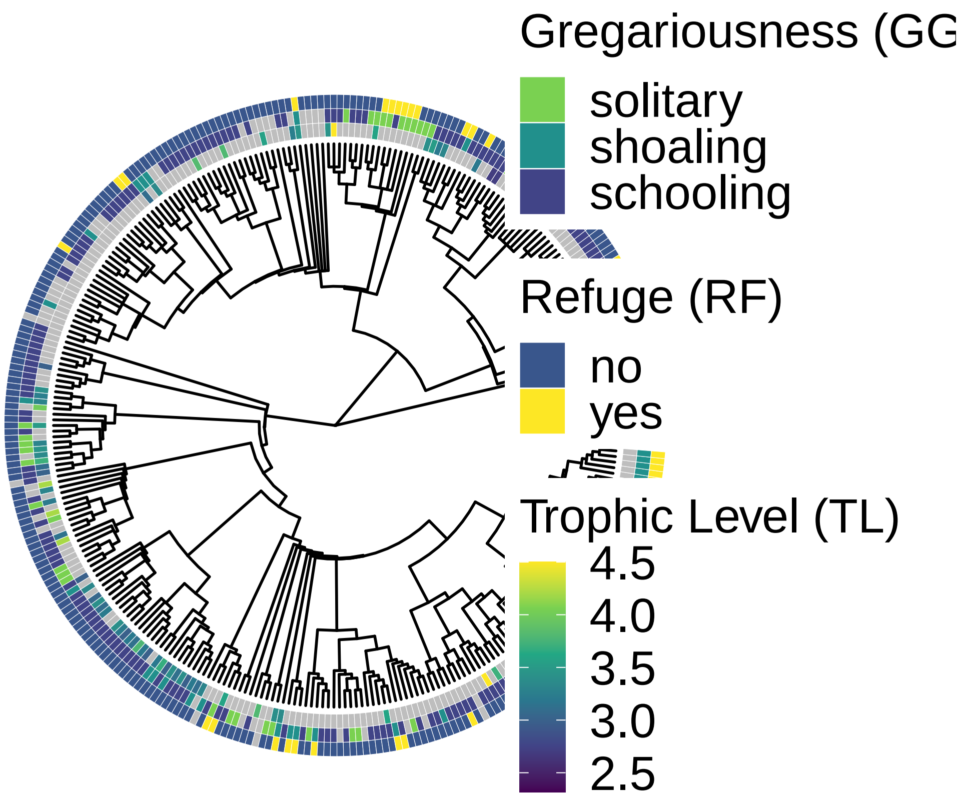
For habitat use traits: we are going to plot multiple rings of data around the phylogenetic trees using vert_habitat, horz_habitat, diel_migrant (change to yes/no), and season_migrant (yes/no)

```
#creating this plot one iteration at a time. Adding one layer of the heatmap, then allowing for another

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
  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"
                       limits = c("reef-associated", "coastal", "continental shelf","continental slope"
                       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], of
  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

```r
levels(as.factor(my_prey_prob$body_shape))
```

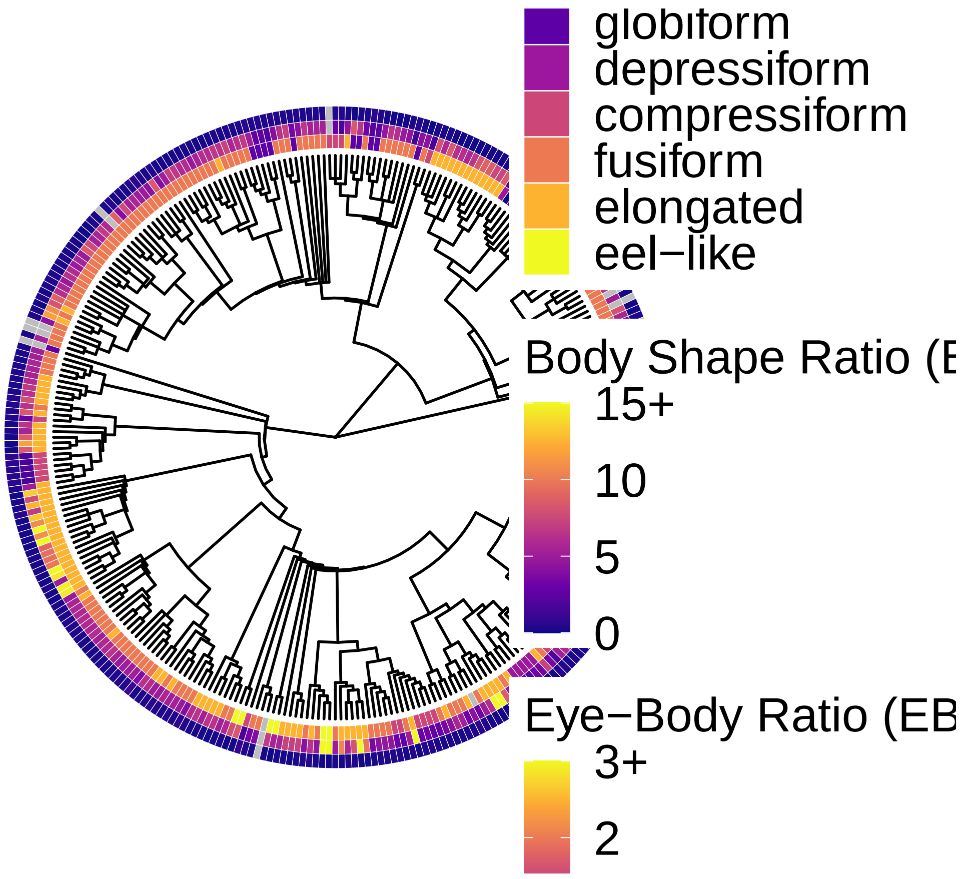
```
## [1] "compressiform" "depressiform"  "eel-like"      "elongated"
## [5] "fusiform"      "globiform"     "unique"
```

```r
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"
```

```r
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()
```

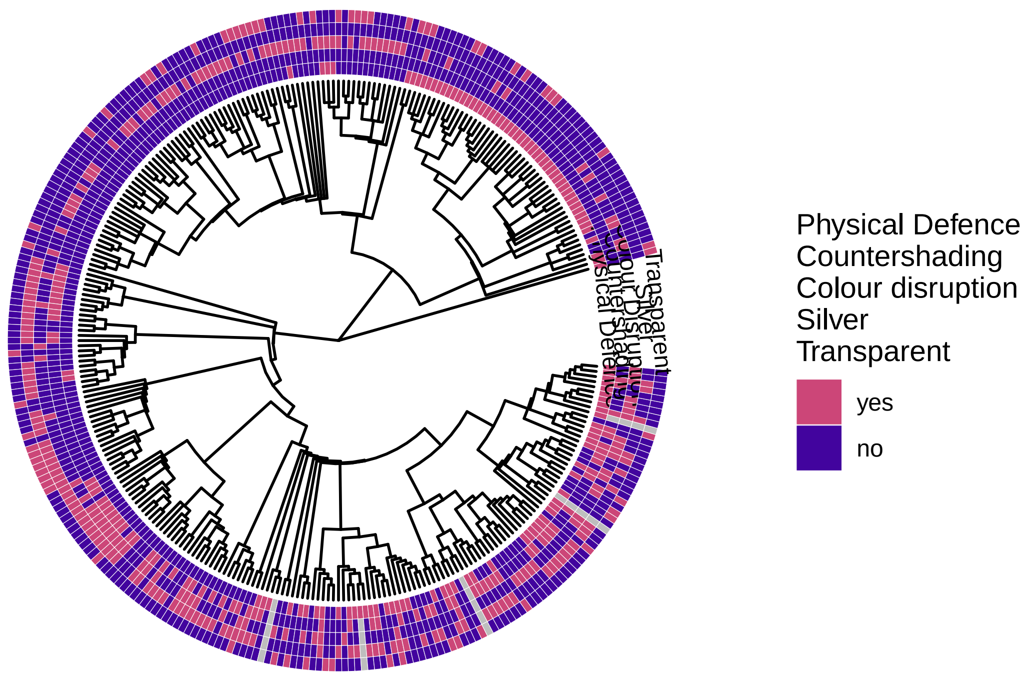
```
morphology_comparison_prob_final <- gheatmap(morphology_comparison_prob2.5,
                                             my_prey_prob[ ,'eye_body_r',drop=FALSE],
                                             offset=0.10, width=0.05, colnames = F) +
  scale_fill_viridis_c(name = "Eye-Body Ratio (EBR)",
                       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)

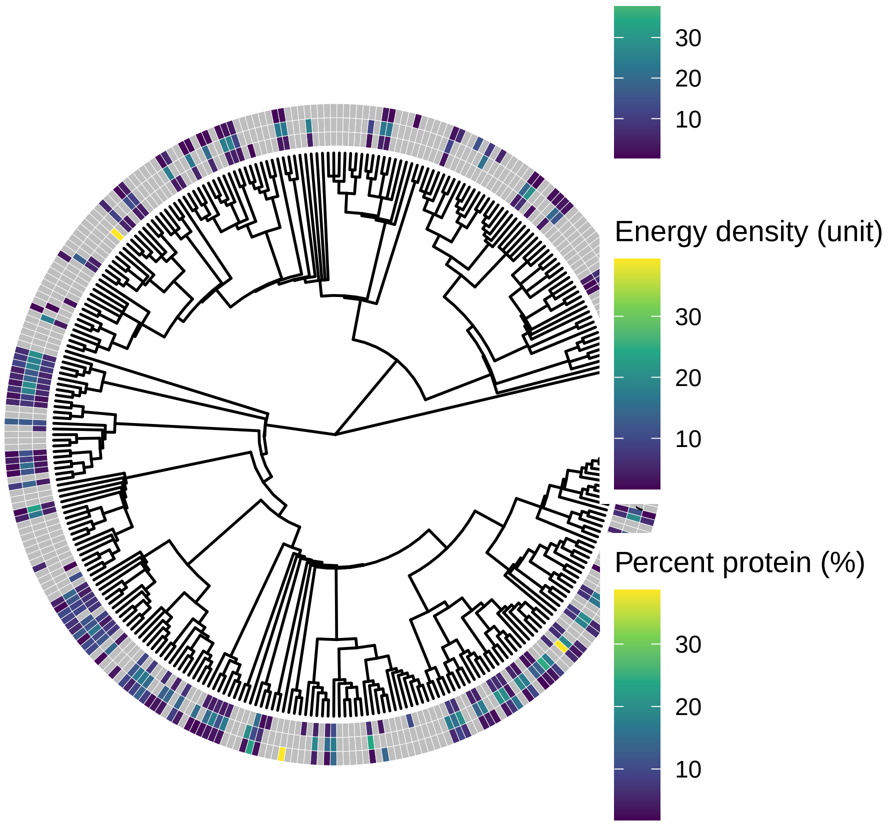
```
morphology_binary_prob1 <- gheatmap(prob_basic, my_prey_prob[ ,'phys_defense',drop=FALSE], offset=0, wi
  scale_fill_viridis_d(name = "Physical Defence", #"\nCountershading\nColour disruption\nSilver\nTransp
                       option = 'C',
                       breaks = c("1", "0"),
                       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\nTransp
                       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\nTransp
                       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", #"\nTransp
                       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=FALS
                                         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
```



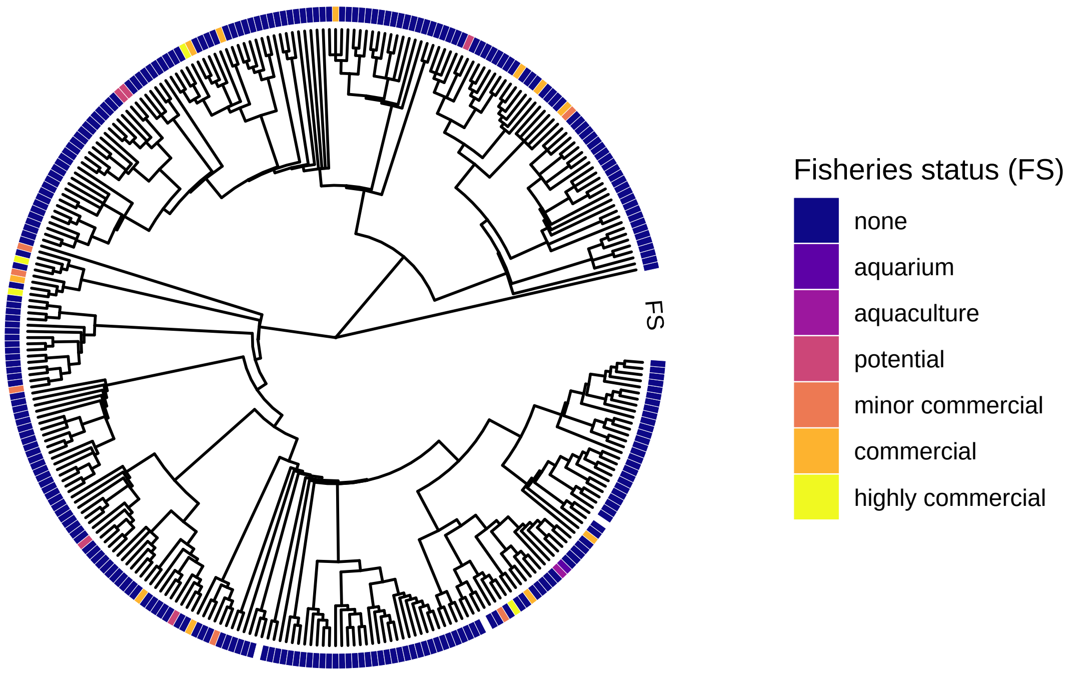
For Prey nutritional quality use: energy_density, percent_protein, percent_lipid

```
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
                                      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',
                                           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