

Albacore Review Trait Data Manipulation

Natasha Hardy

October 10, 2019

About

This document contains code for manipulating trait data sets for albacore prey species from our global meta-analysis of albacore diet composition.

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

```
## [1] "/Users/tashhardy/Documents/GitHub/albacore-diet-global"
```

Outputs inventory

List of outputs created for analyses:

- Prey species' morphometric trait information (body shape, eye diameter to body, and standard to total length): *data/output_data/prey_morph_sum.csv*
- Prey species' nutritional composition traits (energy density, percent lipid and percent protein composition): *data/output_data/prey_qual_sum.csv*
- Prey species' trait values: (i) selected priority trait columns *data/output_data/prey_trait_select.csv* and (ii) trait values for adult life stages only *data/output_data/prey_trait_adult.csv* -> for downstream data joins and analyses.
- Prey species length ratios for max and min length: *data/output_data/prey_length_wide.csv*
- Prey species + filled in length ratios: *data/output_data/prey_length_ratiosp.csv*

Trait Dataframes

Prey morphometric measurements

We first want to load in and clean the prey morphometric measurements for adult taxa only, we did not finish or find many images to work with for juveniles or larvae of most species. The following code aims to:

- Check & select relevant data,
- Calculate eye to body and body shape ratios, based on adult specimen images and morphometrics only.
- Note we have also added a conversion from standard length to total length (this can be used to convert lengths for SL/BL, ML/GL where we have them).
- Summarise to give an average value for each species for which we obtained images. We are able to obtain values for 292 species.

```
#Load prey morphometric measurement data

prey_morph = as.data.frame(read_xlsx(here::here("data/input_data/prey_morphometrics.xlsx"),
                                     sheet = "prey_adult_morphometrics"))

#Convert relevant columns to character/numeric values
prey_morph[,12:17] <- sapply(prey_morph[,12:17], as.numeric) #as.character

#Data cleaning
prey_morph_clean = prey_morph %>%
  dplyr::filter(use %in% c("yes", "maybe")) %>% #c("no", "NA")
  #filter out images that are use = "no"
  drop_na(t_length, b_height, eye_diameter) %>% #filter out any NA's in the desired columns
  mutate(b_shape_r = t_length/b_height,
         eye_body_r = eye_diameter/t_length,
         standard_total = s_length/t_length) #calculate values for body_shape_r & eye_body_r

glimpse(prey_morph_clean)

## Rows: 1,233
## Columns: 24
## $ prey_class      <chr> "Cephalopoda", "Cephalopoda", "Malacostraca", "Actinopt~
## $ prey_family     <chr> "Enoploteuthidae", "Enoploteuthidae", "Alpheidae", "Ste~
## $ prey_sp         <chr> "Abralia redfieldi", "Abraliopsis affinis", "Alpheus gl~
## $ measurer       <chr> "CM", "NA", "IG", "CM", "CM", "CM", "CM", "CM", "CM", "~
## $ file_name       <chr> "Abralia_redfieldi_adult_3.jpg", "Abraliopsis_affinis_a~
## $ use             <chr> "maybe", "yes", "maybe", "maybe", "maybe", "maybe", "ma~
## $ number_reps     <dbl> 4, 1, 6, 6, 1, 1, 1, 6, 2, 2, 5, 5, 5, 1, 4, 4, 1, 3, 3~
## $ source          <chr> "wikipedia; sciencephoto.com; tolweb.org", NA, "J. Leco~
## $ life_stage      <chr> "adult", "adult", "adult", "adult", "adult", "adult", "~
## $ image_type      <chr> "simple sketch", "institute", "amateur", "simple sketch~
## $ scale_bar       <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "N", "~
## $ s_length        <dbl> 140.014, 236.294, 631.165, 520.010, NA, 2275.500, 174.8~
## $ t_length        <dbl> 270.333, 534.294, 695.520, 674.000, 202.540, 2491.400, ~
## $ b_height        <dbl> 59.755, 94.157, 117.720, 362.130, 36.751, 288.560, 25.6~
## $ t_height        <dbl> 122.500, 240.205, 117.720, 471.341, 43.001, 288.560, 41~
```

```
## $ f_height      <dbl> 38.880, 97.473, NA, 65.674, 6.000, 0.000, 8.002, 0.000, ~
## $ eye_diameter  <dbl> 22.000, 40.032, 22.204, 60.008, 0.000, 57.395, 17.334, ~
## $ status_type   <chr> "arm", "NA", "position", "fin", "fin", "fin", "fin", "f~
## $ fin_arm_status <chr> "visible", "NA", "top", "visible", "visible", "absent", ~
## $ annotations   <chr> NA, NA, "Top-Down View --> body height as measured is r~
## $ nh_annotation <chr> "NH checked", "NH measured", "Noted", "NH checked", "NH~
## $ b_shape_r      <dbl> 4.5240231, 5.6745011, 5.9082569, 1.8612101, 5.5111426, ~
## $ eye_body_r     <dbl> 0.08138111, 0.07492504, 0.03192432, 0.08903264, 0.00000~
## $ standard_total <dbl> 0.5179316, 0.4422546, 0.9074721, 0.7715282, NA, 0.91334~
```

Next, we aim to calculate ratios/summarise morphometric data at species resolution for body shape, eye diameter to body, and standard to total length, and we obtain 292 average values out 308 species for which we obtained other trait information:

```
prey_morph_sum = prey_morph_clean %>%
  dplyr::select(pre_y_class:prey_sp, b_shape_r, eye_body_r, standard_total) %>% #life_stage,
  group_by(pre_y_sp) %>% #, life_stage
  dplyr::summarize(b_shape_r = mean(b_shape_r), eye_body_r = mean(eye_body_r),
                  standard_total = mean(standard_total))

#Could calculate standard deviations if needed
#b_shape_se = sd(b_shape_r)/sqrt(length(b_shape_r)), and
#eye_body_se = sd(eye_body_r)/sqrt(length(eye_body_r)).

glimpse(pre_y_morph_sum)
```

```
## Rows: 292
## Columns: 4
## $ prey_sp      <chr> "Abralia redfieldi", "Abraliopsis affinis", "Abraliopsi~
## $ b_shape_r     <dbl> 3.914416, 5.345385, 5.280579, 5.630847, 8.389906, 6.237~
## $ eye_body_r    <dbl> 0.08487754, 0.06979495, 0.05075453, 0.05010374, 0.04213~
## $ standard_total <dbl> 0.5267561, 0.4569214, 0.4583028, 0.4224399, 0.3493369, ~
```

```
write.csv(pre_y_morph_sum, here::here("data/output_data/pre_y_morph_sum.csv"))
```

Prey Quality Data

These include data obtained for prey quality nutritional information for %lipid content, %protein and energy density, typically representing an average value per study for each species of prey, and based on wet weight measured. Note that Zach put together meta-data for this dataframe.

```
#Load prey trait db
prey_qual <- read.csv(here::here("data/input_data/prey_quality_ww.csv"))
str(pre_y_qual) #Check data
```

```
## 'data.frame':    1050 obs. of  9 variables:
## $ ID             : int  1 2 3 4 5 6 7 8 9 10 ...
## $ prey_class     : chr  "Actinopterygii" "Actinopterygii" "Actinopterygii" "Actinopterygii" ...
## $ prey_order     : chr  "Aulopiformes" "Aulopiformes" "Aulopiformes" "Aulopiformes" ...
## $ prey_family    : chr  "Anotopteridae" "Chlorothalmidae" "Chlorothalmidae" "Omusdidae" ...
## $ prey_sp        : chr  "Anotopterus pharao" "Chlorophthalmus agassizi" "Chlorophthalmus agassizi"
```

```
## $ prey_common      : chr  "daggertooth" "shortnose greeneye" "shortnose greeneye" "hammerjaw" ...
## $ energy_density   : num  10.87 NA 4.43 NA 4.3 ...
## $ percent_protein  : num  NA 22.3 NA NA 15.8 ...
## $ percent_lipid    : chr  NA "0.53" NA NA ...

#Correct data type
prey_qual$percent_lipid <- as.numeric(prey_qual$percent_lipid)

#Summarise, we want an average value for each of the three metrics - ED, %L and %P for each species.
prey_qual_sum = prey_qual %>%
  #dplyr::select(prey_class:percent_lipid) %>% #life_stage,
  #drop_na(energy_density:percent_lipid) %>% #filter out any NA's in the desired columns
  group_by(pre_sp) %>% #, life_stage
  dplyr::summarize(energy_density = mean(energy_density, na.rm=TRUE),
                  percent_protein = mean(percent_protein, na.rm=TRUE),
                  percent_lipid = mean(percent_lipid, na.rm=TRUE))

glimpse(pre_qual_sum)

## Rows: 303
## Columns: 4
## $ prey_sp          <chr> "Abralia redfieldi", "Abraliopsis affinis", "Abraliops~
## $ energy_density   <dbl> NaN, NaN, 4.40000, NaN, NaN, NaN, 1.70000, 3.90000, Na~
## $ percent_protein  <dbl> NaN, NaN, 17.40000, NaN, NaN, NaN, NaN, 16.40000, NaN,~
## $ percent_lipid    <dbl> NaN, NaN, NaN, NaN, NaN, NaN, 6.5375, NaN, 1.3000, NaN, NaN~

write.csv(pre_qual_sum, here::here("data/output_data/prey_qual_sum.csv"))
```

Albacore prey trait db

Loading in and selecting key species coarse traits: for habitat association, morphological characteristics and aggregation behaviour. Note that we need to use existing reports or create estimates for prey size and age class in order to proceed in selecting the appropriate trait values for the prey species consumed.

```
# Load prey trait db
prey_trait <- as.data.frame(read_xlsx(here::here("data/input_data/prey_trait_db.xlsx"),
  sheet = "prey_trait_db"))
```

Key traits selection:

```
prey_trait_select = prey_trait %>%
  dplyr::select(prey_class:prey_sp, life_stage, vert_habitat, horz_habitat, diel_migrant,
    diel_migrant_cat, refuge, refuge_cat, season_migrant, season_cat, body_shape,
    l_max, phys_defense, transparent, col_disrupt, silver, countershade, gregarious_primary,
    trophic_level, fisheries_status)

str(prey_trait_select)

## 'data.frame':      855 obs. of  23 variables:
## $ prey_class      : chr  "Hydrozoa" "Malacostraca" "Malacostraca" "Malacostraca" ...
## $ prey_order      : chr  "Siphonophorae" "Decapoda" "Decapoda" "Decapoda" ...
```

```
## $ prey_family      : chr "Diphyidae" "Acanthephyridae" "Acanthephyridae" "Munididae" ...
## $ prey_sp          : chr "Chelophyes appendiculata" "Acanthephyra pelagica" "Acanthephyra pelagica" ...
## $ life_stage       : chr "adult" "adult" "juvenile" "larva" ...
## $ vert_habitat     : chr "epipelagic" "mesopelagic" "mesopelagic" "epipelagic" ...
## $ horz_habitat     : chr "continental shelf" "oceanic" "oceanic" "continental shelf" ...
## $ diel_migrant     : chr "1" "1" "1" "0" ...
## $ diel_migrant_cat : chr "diel_yes" "diel_yes" "diel_yes" "diel_no" ...
## $ refuge           : chr "0" "0" "0" "0" ...
## $ refuge_cat       : chr "refuge_no" "refuge_no" "refuge_no" "refuge_no" ...
## $ season_migrant   : chr "1" "1" "1" "1" ...
## $ season_cat       : chr "season_yes" "season_yes" "season_yes" "season_yes" ...
## $ body_shape       : chr "fusiform" "elongated" "elongated" "globiform" ...
## $ l_max            : num 0.2 27 2.8 0.72 1.7 ...
## $ phys_defense     : num 1 1 1 1 1 1 1 1 1 1 ...
## $ transparent     : chr "1" "0" "0" "1" ...
## $ col_disrupt      : chr "0" "0" "0" "0" ...
## $ silver           : chr "0" "0" "0" "0" ...
## $ countershade     : chr "0" "0" "0" "0" ...
## $ gregarious_primary : chr "schooling" "schooling" "schooling" "schooling" ...
## $ trophic_level    : chr "NA" "NA" "NA" "NA" ...
## $ fisheries_status : chr "none" "none" "none" "none" ...
```

```
write.csv(pre Trait_select, here::here("data/output_data/pre Trait_select.csv"))
```

And select the traits for adult life stage only.

```
pre Trait_adult = pre Trait %>%
  dplyr::filter(life_stage == "adult") %>%
  dplyr::select(pre Trait_class:pre Trait_sp, vert_habitat, horz_habitat, diel_migrant, diel_migrant_cat,
    refuge, refuge_cat, season_migrant, season_cat, body_shape, l_max, phys_defense,
    transparent, col_disrupt, silver, countershade, gregarious_primary, trophic_level,
    fisheries_status)

write.csv(pre Trait_adult, here::here("data/output_data/pre Trait_adult.csv"))
```

Life stage size ratio calculations

In this section of code we calculate size ratios and threshold ratios for larval, juvenile and adult life stages of prey species, through the following:

- Length conversions calculated previously as 'standard_total': to convert any measurements that are not TL into TL.
- Existing lmin and lmax data for each life stage are used to calculate ratios based on existing data;
- Where those data are missing for a species or genus, etc, we calculate those ratios based on data for related species (from genus, family, and so on) to fill in those missing ratios.
- This therefore creates useable ratios for all taxa, using ratios at higher levels of classification when needed.

Manipulate length information

```
# Now need to split the min vs. max lengths to reshape those data and merge
# again later.
prey_min = prey_trait %>%
  dplyr::select(pre_class:prey_sp, life_stage, l_min:l_min_type) %>%
  tidyr::gather(key = "l_min", value = "length", l_min) %>%
  dplyr::rename(measure = l_min, l_type = l_min_type)

prey_max = prey_trait %>%
  dplyr::select(pre_class:prey_sp, life_stage, l_max:l_max_type) %>%
  tidyr::gather(key = "l_max", value = "length", l_max) %>%
  dplyr::rename(measure = l_max, l_type = l_max_type)

# Now we want to bind this data again names(prey_min) #check column names are
# the same for both dfs
prey_length <- rbind(prey_min, prey_max)
```

Here we clean the levels of 'l_type' (length measurement type) factor as follows:

```
prey_length = prey_length %>%
  #Note can use case_when() or if_else() BUT you need to write one for each value
  #and not try to combine them as strings using c()
  mutate(`l_type_use` = case_when(prey_length$l_type == "TL" ~ "TL",
    prey_length$l_type == "SL" ~ "SL",
    prey_length$l_type == "FL" ~ "TL",
    #fork length in some pelagic fishes ~ TL
    prey_length$l_type == "NL" ~ "TL",
    #Notochord length in larvae ~ TL
    prey_length$l_type == "CL" ~ "TL",
    #Carapace length in some crustaceans ~ TL
    prey_length$l_type == "BL" ~ "TL",
    #Body length in larvae ~ TL
    prey_length$l_type == "GL" ~ "SL",
    #Gladius length = Mantle length ~ standard length
    prey_length$l_type == "ML" ~ "SL",
    #Mantle length ~ standard length
    prey_length$l_type == "AN" ~ "TL",
    #Anterior nectophore, appears only in a gastropod,
    #going to take is as TL
    prey_length$l_type == "PCL" ~ "SL"
    #Pre-claval length for sunfish
  )
)
```

NOTE: that we could provide metadata table for measurement type information and/or add it to ESM.

Calculate life stage ratios

Below, we obtain a data set where we have rows for both lmin and lmax for adult, juvenile and larval life stages in columns, as well as the juvenile:adult and larva:adult size ratios where those data exist. We have a number of NAs still, and we will fill those in in the following chunk.

```

# Next we need to merge our ratios data set and our lengths datasets.
prey_length_data = prey_length %>%
  left_join(preymorph_sum, by = "prey_sp") #>% #c('prey_sp', 'life_stage')

prey_length_data = prey_length_data %>%
  mutate(length_tl = if_else(preymorph_sum$l_type_use == "SL", length/standard_total,
    length))

# Reshape to wide data where each species has row for max and min values, and
# add two columns with ratios of larval and juvenile sizes to adult by species.
# We can then summarise these for family, order and class to fill in info.
prey_length_wide = prey_length_data %>%
  dplyr::select(preymorph_sum:life_stage, measure, length_tl) %>%
  spread(life_stage, length_tl, convert = TRUE) %>%
  mutate(larva_adult = 100 * larva/adult, juve_adult = 100 * juvenile/adult)
# we're going to want the maximum values out of this to use as cut-offs

write.csv(preymorph_sum, here::here("data/output_data/preymorph_sum.csv"))

```

Here we create estimate adult:larva and adult:juvenile size ratios based on family, order, class levels of classification. And we will append these to the above dataset where species-level size ratios are not known or length data were incomplete.

```

##Summarise means of larval and juvenile to adult body length ratios, by family, order and class:
#Note that mean() function returns "NaN" whereas sum()/length() returns zeros for missing data.

#By class
preymorph_class <- dplyr::ddply(preymorph_wide, c("preymorph_class", "measure"), summarise,
  larva_adult_c = mean(larva_adult, na.rm=TRUE),
  juve_adult_c = mean(juve_adult, na.rm=TRUE))

#By order
preymorph_order <- dplyr::ddply(preymorph_wide, c("preymorph_order", "measure"), summarise,
  larva_adult_o = mean(larva_adult, na.rm=TRUE),
  juve_adult_o = mean(juve_adult, na.rm=TRUE))

#By family
preymorph_fam <- dplyr::ddply(preymorph_wide,
  c("preymorph_class", "preymorph_order", "preymorph_family", "measure"),
  summarise,
  larva_adult_f = mean(larva_adult, na.rm=TRUE),
  juve_adult_f = mean(juve_adult, na.rm=TRUE))

##Merge these data to calculate prey cut-offs for family, order and class
#--> Code such that we use first family level mean, then order, then class,
# to fill in missing information for prey size cut-offs

preymorph_ratios = preymorph_fam %>%
  left_join(preymorph_order, by= c("preymorph_order", "measure")) %>% #join preymorph_order info
  left_join(preymorph_class, by= c("preymorph_class", "measure")) %>% #join preymorph_class info
  mutate(larva_adult_u = if_else(larva_adult_f == "NaN",
    if_else(larva_adult_o == "NaN", larva_adult_c, larva_adult_o),
    larva_adult_f),

```

```

      juve_adult_u = if_else(juve_adult_f == "NaN",
                           if_else(juve_adult_o == "NaN", juve_adult_c, juve_adult_o),
                           juve_adult_f))
##Make a cut-off selection based on family > order > class level information

#And remove troublesome NA's in original dataset
prey_length_wide$larva_adult[is.na(pre_length_wide$larva_adult)] <- NaN
prey_length_wide$juve_adult[is.na(pre_length_wide$juve_adult)] <- NaN

```

Length ratio data

Finally, the code below provides us with the following data that can be extracted: (i) prey trait data as rows for 3x life stages as columns; (ii) prey length, measurement and type info + average ratio metrics x3 for x3 life stages We can use these to see what essentially fits in albacore jaws, by using gape length and width formulas for yellowfin tuna applied. These are calculated in a later chunk.

```

## Now we need to merge this back onto species, and make selection again based
## on whether the species level info is there

prey_length_ratiosp = prey_length_wide %>%
  dplyr::filter(measure == "l_max") %>%
  dplyr::rename(l_larva = "larva", l_juve = "juvenile", l_adult = "adult") %>%
  left_join(pre_length_ratios, by = c("prey_family", "prey_order", "prey_class",
    "measure")) %>%
  # dplyr::select(pre_class:measure, larva_adult:juve_adult,
  # larva_adult_u:juve_adult_u) %>%
mutate(larva_adult_use = if_else(larva_adult == "NaN", larva_adult_u, larva_adult),
      juve_adult_use = if_else(juve_adult == "NaN", juve_adult_u, juve_adult)) %>%
  dplyr::select(pre_class:l_larva, larva_adult_use:juve_adult_use) %>%
  # dplyr::filter(larva_adult_use != 'NaN')

#---> Here we obtain size ratios for all 309 species in trait db
## Problem with NA values
glimpse(pre_length_ratiosp)

```

```

## Rows: 309
## Columns: 10
## $ prey_class      <chr> "Actinopterygii", "Actinopterygii", "Actinopterygii", ~
## $ prey_order      <chr> "Argentiniformes", "Argentiniformes", "Ateleopodiforme~
## $ prey_family     <chr> "Bathylagidae", "Bathylagidae", "Ateleopodidae", "Athe~
## $ prey_sp         <chr> "Bathylagoides wesethi", "Leuroglossus stilbius", "Ate~
## $ measure         <chr> "l_max", "l_max", "l_max", "l_max", "l_max", "l_max", ~
## $ l_adult         <dbl> 15.71274, 15.00000, 60.00000, 45.00000, 215.00000, 146~
## $ l_juve          <dbl> NA, NA, 25.80000, 16.00000, NA, 43.00000, 81.80303, 11~
## $ l_larva         <dbl> 2.500000, 3.114180, NA, 2.362748, NA, NA, 5.388869, 3.~
## $ larva_adult_use <dbl> 15.910658, 20.761202, Inf, 5.250552, 10.405454, 5.2083~
## $ juve_adult_use  <dbl> 42.11736, 42.11736, 43.00000, 35.55556, 40.22302, 29.4~

```

```

write.csv(pre_length_ratiosp, here::here("data/output_data/prey_length_ratiosp.csv"))

```