Ecostar WP4 - Data Preparation

Clement Garcia

2024-03-15

## Formatting biological data

The original source of the biological data are stored in data\_raw, from which the Essex team (UoE) did a thorough clean, standardisation and various formatting, the details of which are fully explained in the “data\_processed/README.md”.

## Adding Juvenile data \* not to be done again  
nameCor<-read.csv("./Name\_Correction.csv")  
SITE<-UKB\_speMaster %>%   
 select(NO,SITE) %>%   
 distinct(SITE, .keep\_all=T)  
  
check<-left\_join(SITE, nameCor)  
check2<-check[grep("(juv)", check$SPECIES), c("SITE", "SPECIES")]  
#write.csv(check2, "./Essex/Site\_Check.csv")  
  
siteChck<-read.csv("./Site\_Check.csv")  
siteChckName<-read.csv("./SiteCheckName.csv")  
siteChckName$lifeCor<-"juvenile"  
temps<-left\_join(siteChck, siteChckName)  
temps<-temps %>%   
 distinct(SITE, NEW\_SPECIES, lifeCor)  
  
  
## Formatting  
speM<-UKB\_speMaster[, c("new\_species", "habitat", "habitat\_class", "feeding\_strat")]  
speM<-left\_join(speM, UKB\_speList)  
speM<-speM[, c("new\_species", "acceptedName", "aphiaID", "kingdom", "phylum", "class", "order",  
 "family", "genus", "subgenus", "species", "speciesDetails", "habitat", "habitat\_class", "feeding\_strat")]  
  
speSite<-UKB\_Data[, c("SITE", "SPECIES", "abundance", "SURVEY", "YEAR", "MONTH", "STAT\_NO", "SEASON")]  
colnames(speSite)<-c("site", "new\_species", "abn", "survey", "year", "month", "stat\_no", "season")  
  
spe<-left\_join(speSite, speM)  
spe<-spe %>%   
 group\_by(site, survey, year, month, stat\_no, season, acceptedName, aphiaID, kingdom, phylum, class, order, family,  
 genus, subgenus, species, speciesDetails, habitat, habitat\_class, feeding\_strat) %>%   
 summarise(A = mean(abn, na.rm=T))  
  
  
spe<-left\_join(spe, temps, by = c("site" = "SITE", "acceptedName" = "NEW\_SPECIES"))  
  
spePlat<-spe[!spe$lifeCor %in% "juvenile",]  
spePlat<-spe[!is.na(spe$acceptedName),]  
spePlat<-as.data.frame(spePlat)  
spePlat$taxoLevel = with(spePlat,  
 ifelse(!is.na(species), "species",  
 ifelse(!is.na(genus), "genus",  
 ifelse(!is.na(family), "family",  
 ifelse(!is.na(order), "order",  
 ifelse(!is.na(class), "class",  
 ifelse(!is.na(phylum), "phylum",  
 ifelse(!is.na(kingdom), "kingdom", NA))))))))  
  
  
## Add average biomass  
bMass<-read.csv("./body\_mass\_gWW\_FINAL.csv")  
  
##\*Body Mass matching  
#Unique taxa record from main database  
taxa\_unique<-spePlat %>%   
 distinct(acceptedName, kingdom, phylum, class, order, family, genus, species, taxoLevel)  
colnames(taxa\_unique)[1]<-"ScientificName\_accepted"  
#Join the taxa with their respective recorded body mass  
raw\_gww<-left\_join(taxa\_unique, bMass[bMass$lvlEstimate %in% "taxa",], by = "ScientificName\_accepted")  
## Keep what has a direct match  
spe\_gww<-raw\_gww[!is.na(raw\_gww$Meanbodymass),]  
spe\_gww$taxoMassLvl<-"asRecorded"  
## First leftovers  
speNA<-raw\_gww[is.na(raw\_gww$Meanbodymass),]  
## Species or Genus (to avoid the NA match)  
gen\_temp<-speNA[speNA$taxoLevel %in% c("species", "genus"),]  
## Second left over (first part) what is not species or genus  
genNA<-speNA[!speNA$taxoLevel %in% c("species", "genus"),]  
## Join the taxa with match at Genus level  
gen\_gww<-left\_join(gen\_temp[, -which(names(gen\_temp) %in% c("Meanbodymass","lvlEstimate"))],   
 bMass[bMass$lvlEstimate %in% "genus",],   
 by = c("genus" = "ScientificName\_accepted"))  
## Second left over (second part) no match added to the first part  
genNA<-rbind(genNA, gen\_gww[is.na(gen\_gww$Meanbodymass),])  
## Keep what has a Genus match  
gen\_gww<-gen\_gww[!is.na(gen\_gww$Meanbodymass),]  
gen\_gww$taxoMassLvl<-"asGenus"  
## Species, Genus or Family (to avoid the NA match)  
fam\_temp<-genNA[genNA$taxoLevel %in% c("species", "genus", "family"),]  
## Third left over (first part) what is not species or genus or family  
famNA<-genNA[!genNA$taxoLevel %in% c("species", "genus", "family"),]  
## Join the taxa with match at Family level  
fam\_gww<-left\_join(fam\_temp[, -which(names(fam\_temp) %in% c("Meanbodymass","lvlEstimate"))],   
 bMass[bMass$lvlEstimate %in% "family",],   
 by = c("family" = "ScientificName\_accepted"))  
## Second left over (second part) no match added to the first part  
famNA<-rbind(famNA, fam\_gww[is.na(fam\_gww$Meanbodymass),])  
## Keep what has a Family match  
fam\_gww<-fam\_gww[!is.na(fam\_gww$Meanbodymass),]  
fam\_gww$taxoMassLvl<-"asFamily"  
## Species, Genus, Family, Order (to avoid the NA match)  
ord\_temp<-famNA[famNA$taxoLevel %in% c("species", "genus", "family", "order") & !is.na(famNA$order),]  
## Fourth left over (first part) what is not species or genus or family or Order  
ordNA<-famNA[is.na(famNA$order),]  
## Join the taxa with match at Order level  
ord\_gww<-left\_join(ord\_temp[, -which(names(ord\_temp) %in% c("Meanbodymass","lvlEstimate"))],   
 bMass[bMass$lvlEstimate %in% "order",],   
 by = c("order" = "ScientificName\_accepted"))  
## Second left over (second part) no match added to the first part  
ordNA<-rbind(ordNA, ord\_gww[is.na(ord\_gww$Meanbodymass),])  
## Keep what has a Order match  
ord\_gww<-ord\_gww[!is.na(ord\_gww$Meanbodymass),]  
ord\_gww$taxoMassLvl<-"asOrder"  
## Species, Genus, Family, Order, Class (to avoid the NA match)  
cla\_temp<-ordNA[ordNA$taxoLevel %in% c("species", "genus", "family", "order", "class") & !is.na(ordNA$class),]  
## Fifth left over (first part) what is not species or genus or family or Order or Class  
claNA<-ordNA[is.na(ordNA$class),]  
## Join the taxa with match at Class level  
cla\_gww<-left\_join(cla\_temp[, -which(names(cla\_temp) %in% c("Meanbodymass","lvlEstimate"))],   
 bMass[bMass$lvlEstimate %in% "class",],   
 by = c("class" = "ScientificName\_accepted"))  
## Second left over (second part) no match added to the first part  
claNA<-rbind(claNA, cla\_gww[is.na(cla\_gww$Meanbodymass),])  
## Keep what has a Class match  
cla\_gww<-cla\_gww[!is.na(cla\_gww$Meanbodymass),]  
cla\_gww$taxoMassLvl<-"asClass"  
claNA$taxoMassLvl<-"noInfo"  
###  
#Final compilation  
final<-rbind(spe\_gww, gen\_gww, fam\_gww, ord\_gww, cla\_gww, claNA)  
  
  
## Binding average body mass to the original data  
body\_mass<-final[, c("ScientificName\_accepted", "lvlEstimate", "taxoMassLvl", "Meanbodymass")]  
  
spePlat<-left\_join(spePlat, body\_mass, by = c("acceptedName" = "ScientificName\_accepted"))  
#check<-unique(spePlat2[is.na(spePlat2$Meanbodymass), "acceptedName"])  
#taxa\_unique[taxa\_unique$ScientificName\_accepted %in% check,]  
#missing Foraminifera, Platyhelminthes, Entoprocta, Porifera, Ciliophora, & high level taxo - OK  
  
#save here  
#save(spePlat, file="C:/Users/cg05/OneDrive - CEFAS/Science/Project - Commercial/INSITE/INSITE II/analysis/data/UKBenthos\_spePlat.RData")

## Extracting environmental data from available raster

There was not much environmental information in the original dataset, therefore we used available data from raster (Mitchell et al. 2019) was used in conjonction with platform coordinates to extract the relevant data.



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.