

# Phylogeny\_\_construction

*Kevin Healy*

*20/05/2019*

This script builds a set of 100 phylogenies by joining various phylogenies together and adding error at each node based on the first appearance in fossil evidence. This set of phylogenies is then used for the subsequent analysis in the paper Healy et al. 2019.

## Packages and Data

First lets load the packages. Apart from `mulTree`, which is available on github, all other packages are on CRAN.

`mulTree` is a package designed to run multiple phylogenies in the `MCMCglmm` function and we will use the `multiphylo` format to store our phylogenies for later analysis. It is available to download from <https://github.com/TGuillerme/mulTree>

```
require(mulTree)
```

```
## Loading required package: mulTree
```

```
## Loading required package: ape
```

```
## Loading required package: coda
```

```
## Loading required package: hrcde
```

```
## This is hrcde 3.3
```

```
## Loading required package: MCMCglmm
```

```
## Loading required package: Matrix
```

```
## Loading required package: snow
```

```
library(devtools)
```

```
library(MASS)
```

```
library(popbio)
```

```
library(popdemo)
```

```
## Welcome to popdemo! This is version 1.3-0
```

```
## Use ?popdemo for an intro, or browseVignettes('popdemo') for vignettes
```

```
## Citation for popdemo is here: doi.org/10.1111/j.2041-210X.2012.00222.x
```

```
## Development and legacy versions are here: github.com/iaimstott/popdemo
```

```
library(phytools)
```

```
## Loading required package: maps
```

```
## Loading required package: rgl
```

```
##
```

```
## Attaching package: 'phytools'
```

```
## The following object is masked from 'package:Matrix':
```

```
##
```

```
##      expm
```

Lets also load some functions that we built specifically for this analysis The Demography\_functions script has functions related to calculating demography elements while phylo\_bind\_functions script has functions to build the phylogenies and include error in the nodes joining together trees. These scripts both available on the Healy\_et\_al\_2019\_Animal\_Life\_History github repository.

```
source("Demography_functions.R")
```

```
## Loading required package: mvtnorm
```

```
source("phylo_bind_functions.R")
```

Now lets load the demography dataset so that we can match up our species with the phylogeny. For the paper we used a version of COMADRE which was released with the publication and is available on this github page. However, check <https://www.compadre-db.org/Data/Comadre> for potential updates.

```
load("COMADRE_v_2.0.0.9.RData")
```

Lets now upload each of the phylogenies we are going to use to build our super-doooper tree. We used the Metazoan phylogeny from the Open Tree of Life project (<https://ot39.opentreeoflife.org/opentree/argus/opentree10.4@ott93302>) as a backbone. For Mammals we used a distribution of 100 trees from Kuhn et al 2011. For Aves we used the Jetz et al 2012 distribution of trees. For lizards we used the Pyron et al 2014 phylogeny.

For our analysis we will use a distribution of 100 supertrees, however as many of the trees will not be suitable (for example, when first apperance fossil ranges overlap between nodes resulting in negative distances for certain topologies.)

\*Please note that as the seperate phylogenies are too large to host on Github they will need to be downloaded seperately from there sources. However, we do include the output distribution of trees (COMADRE\_100\_phylo\_feb\_2019.tre) that is the result of this script within the repository.

```
#Metazoan backbone
```

```
met_tree <- read.tree("metazoa.tre")
```

```
met_tree <- makeLabel(met_tree)
```

```
#Mammal phylogeny
```

```
mammal_phy <- read.nexus("FritzTree.rs200k.100trees.tre")
```

```
#Aves phylogeny
```

```
aves_phy <- read.tree("BirdzillaHackett10.tre")
```

```
#Lizard phylogeny
```

```
Lizard_phy <- read.tree("liz_and_snake_time_tree.txt")
```

```
lizard_phy<-makeLabel(Lizard_phy)
```

```
# lets also set the number of trees we will build.
```

```
# As we will have to drop trees due to the inclusion of uncertinatlly in node age
```

```
# creating trees that dont correspond with the structure of our backbone phylogeny
```

```
# we need to more than 100
```

```
sample.no <- 300
```

## COMADRE species list

To get our list of species that we will build the tree for we use the same subsetting criteria used in the main analysis (See paper for more details).

```

pooled_Metadata <- (subset(comadre$metadata,
                          MatrixComposite == "Pooled"
                          & MatrixDimension >= 2
                          & StudyDuration >= 2
                          & MatrixSplit == "Divided"
                          & MatrixFec == "Yes"
                          & MatrixTreatment == "Unmanipulated"
                          & AnnualPeriodicity == "1"
))

mean_Metadata <- (subset(comadre$metadata,
                          MatrixComposite == "Mean"
                          & MatrixDimension >= 2
                          & StudyDuration >= 2
                          & MatrixSplit == "Divided"
                          & MatrixFec == "Yes"
                          & MatrixTreatment == "Unmanipulated"
                          & AnnualPeriodicity == "1"
))

combined_data <- rbind(pooled_Metadata, mean_Metadata )
keep_first <- as.numeric(rownames(combined_data))

species_list <- data.frame(species =(combined_data$SpeciesAccepted),
                          class = (combined_data$Class),
                          phyla = (combined_data$Phylum), stringsAsFactors=FALSE )
#add humans
human_list <- c(species = "Homo_sapiens", class = "Mammalia", phyla = "Chordata" )
species_list <- rbind(species_list, human_list)

species_list_u <- unique(species_list$species)

clean_species <- comadre$metadata[keep_first,]$SpeciesAccepted
species_clean_list_u <- unique(clean_species)

full_uni <- data.frame(species_grep = species_clean_list_u,
                      species = species_clean_list_u ,
                      species_match = rep(0, length(unique(species_clean_list_u))),
                      matched = rep(0, length(unique(species_clean_list_u))),
                      taxonomic_class = rep(0, length(unique(species_clean_list_u))),
                      stringsAsFactors=FALSE)

##add_humans
human_list_u <- c(species_grep = "Homo_sapiens",
                  species = "Homo_sapiens", 0,0,0)

full_uni <- rbind(full_uni,
                  human_list_u)

```

We then clean addition text and change the species names format from “Genus species” to “Genus\_species” so they columnes matched between the data and the phylogenies.

```
###remove all the subspecies information
full_uni[,1] <- gsub(" subsp.*","", full_uni[,1])
full_uni[,1] <- gsub(" ","_", full_uni[,1])

species_list[,1] <- gsub(" subsp.*","", species_list[,1])
species_list[,1] <- gsub(" ","_", species_list[,1])
```

We now match up the species names from COMADRE with those in the backbone tree

*#This pulls out all the entries in the open tree of life phylogeny that correspond  
#with the species names in our COMADRE species list.*

```
for(i in 1:(length(full_uni[,1]))) {
  if(any(grep(full_uni[i,1], met_tree$tip.label)) == T) {
    full_uni[i,3] <- met_tree$tip.label[grep(full_uni[i,1], met_tree$tip.label)][1]
    full_uni[i,4] <- "yes"
  }

  else {
    full_uni[i,3] <- full_uni[i,1]
    full_uni[i,4] <- "no"
    full_uni[i,5] <- species_list[species_list$speciesAcc == c(full_uni[i,2]), "class"][1]
  }
}

###addition of Class information
for(i in 1:(length(full_uni[,1]))) {

  full_uni[i,5] <- as.vector(species_list[(grep(full_uni[i,1], species_list[,1]))[1], "class"])
}

####remove any duplicates that might have arise from sub-species
full_uni <- full_uni[which(duplicated(full_uni$species_match) == FALSE),]

full_uni <- data.frame(full_uni,
                      species_two = rep(0, length(full_uni[,1])))
```

Now we match up the species from the taxonomic groups in our dataset which we do not have a dated phylogeny.

###for taxonomic groups without a dated tree we need to handle them more directly

```
taxa_ott_match <- full_uni[full_uni$taxonomic_class %in% c("Actinopterygii",
                                                         "Amphibia",
                                                         "Anthozoa",
                                                         "Bivalvia",
                                                         "Insecta",
                                                         "Demospongiae",
                                                         "Elasmobranchii",
                                                         "Gastropoda",
                                                         "Malacostraca"),

                          "species_match"]
```

```

###we also need to make a list of turtle and croc species

turt_names <- c("Caretta_caretta_ott392505",
               "Chelodina_expansa_ott642964",
               "Chelydra_serpentina_ott587311",
               "Chrysemys_picta_marginata_ott801475",
               "Clemmys_guttata_ott436621",
               "Emydura_macquarii_krefftii_ott687926",
               "Kinosternon_subrubrum_steindachneri_ott5223436",
               "Malaclemys_terrapi_terrapi_ott227420",
               "Podocnemis_expansa_ott709468",
               "Sternotherus_odoratus_ott487672",
               "Kinosternon_integrum_ott944915",
               "Kinosternon_flavescens_spooneri_ott487666",
               "Podocnemis_lewyana_ott211855")

croc_names <- c("Crocodylus_johnsoni_ott458978",
               "Crocodylus_niloticus_ott35864")

###change the names for the species within reptiles (i.e turtles)
species_ott_match <- full_uni[full_uni$species_match %in% turt_names,
                             "species_match"]

species_ott_match_croc <- full_uni[full_uni$species_match %in% croc_names,
                                  "species_match"]

tot_ott <- unlist(list(taxa_ott_match,
                     species_ott_match,
                     species_ott_match_croc))

species_two <- vector()

###this re-reads the above species back into a list of matched names
for(i in 1:(length(full_uni$species_match))){
  if(any(full_uni$species_match[i] == tot_ott) == T) {
    species_two[i] <- full_uni$species_match[i]
  } else{

    species_two[i] <- as.vector(full_uni$species_grep[i])

  }
}

```

Next we deal with the synonyms so that the data and trees match up.

```

#####The following species need to be named to match the phylogenies.
###changes for Jetz
species_two[species_two == "Thalassarche_melanophrys"] <- "Thalassarche_melanophrys"
species_two[species_two == "Anser_caerulescens"] <- "Chen_caerulescens"
species_two[species_two == "Anthropoides_paradiseus"] <- "Grus_paradisea"
species_two[species_two == "Setophaga_cerulea"] <- "Dendroica_cerulea"

```

```

species_two[species_two == "Sternula_antillarum"] <- "Sterna_antillarum"

####changes for Prion phylgeny
species_two[species_two == "Cryptophis_nigrescens"] <- "Rhinoplocephalus_nigrescens"
species_two[species_two == "Drymarchon_couperi"] <- "Drymarchon_corais"

species_two[species_two == "Hoplocephalus_bungaroides"] <- "Hoplocephalus_bitorquatus"

####changes for mammal phylogeny
species_two[species_two == "Callospermophilus_lateralis"] <- "Spermophilus_lateralis"
species_two[species_two == "Urocitellus_columbianus"] <- "Spermophilus_columbianus"
species_two[species_two == "Urocitellus_armatus"] <- "Spermophilus_armatus"
species_two[species_two == "Urocitellus_beldingi"] <- "Spermophilus_beldingi"
species_two[species_two == "Physeter_macrocephalus"] <- "Physeter_catodon"
species_two[species_two == "Antechinus_agilis"] <- "Antechinus_stuartii"

full_uni$species_two <- species_two

```

We now start with aves and work backward adding each taxonomic groups and there phylogeny as we meet them. First up is the common ancestor between the two crocodiles species in our analysis

We scale the tree using the Grafen transformation to reflect the divergence time for the last common ancestor between our Croc species which is between 10.77mya and 16.7mya. See Srikulnath, K; Thapana, W; Muangmai, N (2015). "Role of chromosome changes in Crocodylus evolution and diversity". Genomics Inform. 13 (4): 102–111.

```

full_mcl_croc <- full_uni[(full_uni$species_two %in% cror_names),]

fin_croc <- comparative.data(phy = met_tree,
                             data = full_mcl_croc,
                             names.col = "species_match" ,
                             force.root = TRUE)

tree_croc <- compute.brlen(fin_croc $phy,
                           method = "Grafen",
                           power = 1)

tree_croc$edge.length <- tree_croc$edge.length*13.735

```

We now graft these two species onto the aves tree with error using our jiggle.bind function and the range of common ancestor between these two groups as 186mya and 249mya from Chiari et al 2012

```

croc_bird_o <- jiggle.bind(x = tree_croc,
                           y = aves_phy,
                           sample = sample.no,
                           min.age = 186,
                           max.age = 249)

croc_bird <- croc_bird_o[[1]]

croc_bird_ages <- croc_bird_o[[2]]

```

We next graft the composite aves and crocodile (Archosauria) tree onto the turtles. We use a split between

the Pleurodira and the Cryptodira as 157mya from Chiari et al 2012 with the common ancestor between Archosauria and turtles between 274-233mya (<http://onlinelibrary.wiley.com/doi/10.1111/ede.12081/full>)

```
###now add turtles
full_mcl_turt <- full_uni[(full_uni$species_two %in% turt_names),]

fin_turt <- comparative.data(phy = met_tree,
                             data = full_mcl_turt,
                             names.col = "species_match" ,
                             force.root = TRUE)

tree_turt <- compute.brlen(fin_turt $phy,
                           method = "Grafen",
                           power = 1)

tree_turt$edge.length <- tree_turt$edge.length*157

turt_bird_o <- jiggle.bind(x = croc_bird,
                           y = tree_turt,
                           sample = sample.no,
                           min.age = 233,
                           max.age = 274)

turt_bird <- turt_bird_o[[1]]
turt_bird_ages <- turt_bird_o[[2]]
```

To link these trees to the common ancestor of Lepidosauria we used a range of 259-285 Myr from Jones et al 2013

```
liz_turt_bird_o <- jiggle.bind(x = lizard_phy,
                              y = turt_bird,
                              sample = sample.no,
                              min.age = 259,
                              max.age = 285)

liz_turt_bird <- liz_turt_bird_o[[1]]

liz_turt_bird_ages <- liz_turt_bird_o[[2]]
```

For the common ancestor of amniotes, we used the fossil Archerpeton anthracos (Holotype: RM 12056, Author: Carroll 1964, Reisz and Müller, Epoch: Westphalian A Canada Nova Scotia, Age: 318.1 – 314.6 Myr, Dating: International Commission on Stratigraphy 2009)

```
amniote_tree_o <- jiggle.bind(x = liz_turt_bird,
                              y = mammal_phy,
                              sample = sample.no,
                              min.age = 314.6,
                              max.age = 318.1)

amniote_tree <- amniote_tree_o[[1]]

amniote_tree_ages <- amniote_tree_o[[2]]
```

The common ancestor of Actinopterygians and Sarcopterygians was dated using the fossil Guiyu oneiros (Holotype: IVPP V15541, Zhu et al 2013, Epoch: Late Ludlow, Silurian, Kuantu Formation; Qujing, Yunnan,

China., Age: 419 Myr). For the common ancestor between Osteichthyans and Chondrichthyes we used the fossil *Entelognathus primordialis* (Holotype: IVPP V18620, Zhu et al 2009, Epoch: Xiaoxiang Reservoir, Qujing, Yunnan, China Kuantu Formation. Age: 419).

```
fish_data <- full_uni[full_uni$taxonomic_class %in% c("Actinopterygii"),]

fish_tree <- comparative.data(phy = met_tree,
                             data = fish_data,
                             names.col = "species_match" ,
                             force.root = TRUE)

tree_fish <-compute.brlen(fish_tree$phy,
                          method = "Grafen",
                          power = 1)

##Need to get a sensible split. age should be at lungfish or something
tree_fish $edge.length <- tree_fish $edge.length*418.5

fish_tetrapod_tree_o <- jiggle.bind(x = amniote_tree,
                                   y = tree_fish,
                                   sample = sample.no,
                                   min.age = 419,
                                   max.age = 419)

fish_tetrapod_tree <- fish_tetrapod_tree_o[[1]]

fish_tetrapod_tree_ages <- fish_tetrapod_tree_o[[2]]

##I need to add in the one Elasmobranch
shark_names <- full_uni[full_uni$taxonomic_class %in% c("Elasmobranchii"),"species_two"]

deuterostome_tree <- bind_single(tree = fish_tetrapod_tree,
                                 species = shark_names,
                                 sample = sample.no,
                                 root.age = 420)
```

For the common ancestor between deuterostomes and protostomes we use the date of 555-558 Myr for the fossil *Kimberella quadrata* (Holotype: PI 2734, Fedonkin et al 2007, Epoch: Ediacarn, Formation; South Australia).

```
proto_animals <- full_uni[full_uni$taxonomic_class %in% c("Bivalvia",
                                                         "Gastropoda",
                                                         "Insecta",
                                                         "Malacostraca"),]

bi_tree <- comparative.data(phy = met_tree,
                           data = proto_animals,
                           names.col = "species_match",
                           force.root = TRUE)

bi_tree <-compute.brlen(bi_tree$phy,
                       method = "Grafen",
                       power = 1)
```



```

bi_tree $edge.length <- bi_tree $edge.length*550

bilateral_tree_o <- jiggle.bind(x = deuterostome_tree,
                              y = bi_tree,
                              sample = sample.no,
                              min.age = 555,
                              max.age = 558)

bilateral_tree <- bilateral_tree_o[[1]]

bilateral_tree_ages <- bilateral_tree_o[[2]]

```

The common ancestor used for Bilateria and Cnidaria was based on the Haootia fossil 560mya (Holotype: PI 2734, Fedonkin et al 2007, Epoch: Ediacarn, Formation; South Australia). Finally, the age of 800 Myr was used for the common ancestor between Porifera and metazoa was from Erwin et al 2011.

```

##I need to add in the one Antozoa
coral_names <- full_uni[full_uni$taxonomic_class %in% c("Anthozoa"),
                      "species_two"]

two_tissue_tree <- bind_single(tree = bilateral_tree,
                              species = coral_names,
                              sample = sample.no,
                              root.age = 560)

##I need to add in the one sponge
sponge_names <- full_uni[full_uni$taxonomic_class %in% c("Demospongiae"),
                      "species_two"]

animal_tree <- bind_single(tree = two_tissue_tree,
                          species = sponge_names,
                          sample = sample.no,
                          root.age = 800)

```

Next we check if the trees are behaving with no jumping forward in time within the nodes.

```

final_tree<- list()
for(i in 1:(length(animal_tree))){
  animal_tree[[i]]$node.label = "NA"
}

##Now we clean each of the trees so it only has COMADRE species in it.
for(i in 1:(length(animal_tree))){
  final_tree[[i]] <- comparative.data(phy = animal_tree[[i]],
                                     data = full_uni,
                                     names.col = "species_two",
                                     force.root = TRUE)$phy
}

###check that the ancestral nodes dont jump forward in time
###first put the ages with youngest in row 1 an doldest in the last row
node_bind_ages <- rbind(croc_bird_ages,
                       turt_bird_ages,

```

```

        liz_turt_bird_ages,
        amniote_tree_ages,
        fish_tetrapod_tree_ages,
        bilateral_tree_ages)

###check each column is monotonously increasing
mono_tim <- vector()
for(i in 1:sample.no){
  mono_tim[i] <- all(node_bind_ages[,i] == cummax(node_bind_ages[,i]))
}

mono_tim <- data.frame(mono_tim, row_names = 1:sample.no)
mono_clean <- mono_tim[mono_tim[,1] == TRUE, "row_names"]

##now make a list of the clean trees
final_clean_trees <- list()
for(i in 1:length(mono_clean)){
  final_clean_trees[[i]] <- final_tree[[mono_clean[i]]]
}

```

Finally we rename the tips so that they will match the COMADRE species names for the rest of the analysis

```

##now lets rename the tips back to match the COMADRE species
for(k in 1:length(mono_clean)){
  renamed_temp <- final_clean_trees[[k]]

  for(i in 1:(length(renamed_temp$tip.label))){

    if(any(renamed_temp$tip.label[i] == full_uni$species_match)){

      final_clean_trees[[k]]$tip.label[i] <- gsub(" ",
                                                    "-",
                                                    full_uni[renamed_temp$tip.label[i] ==
                                                                full_uni$species_match,
                                                                "species"])
    }
  }
}

for(k in 1:length(final_clean_trees)){

  final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
    == "Thalassarche_melanophrys"] <- "Thalassarche_melanophris"

  final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
    == "Physeter_catodon"] <- "Physeter_macrocephalus"

  final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
    == "Spermophilus_armatus"] <- "Urocitellus_armatus"
}

```

```

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Spermophilus_columbianus"] <- "Urocitellus_columbianus"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Grus_paradisea"] <- "Anthropoides_paradiseus"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Dendroica_cerulea"] <- "Setophaga_cerulea"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Sterna_antillarum"] <- "Sternula_antillarum"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Rhinoplocephalus_nigrescens"] <- "Cryptophis_nigrescens"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Drymarchon_corais"] <- "Drymarchon_couperi"

final_clean_trees[[k]]$tip.label[final_clean_trees[[k]]$tip.label
  == "Hoplocephalus_bitorquatus"] <- "Hoplocephalus_bungaroides"

}

#small fix on ultrametric problem
is_ultra <- vector()
for(i in 1:length(final_clean_trees)) {
  is_ultra[i] <- is.ultrametric(final_clean_trees[[i]])
}

ultra_row <- which(is_ultra ==TRUE)

final_clean_trees_ultra <- final_clean_trees[ultra_row]

##lets take the first 100
final_clean_trees <- final_clean_trees_ultra[1:100]

```

Last we read out our lovely clean multiphylo object.

```

class(final_clean_trees) <- "multiPhylo"

write.tree(phy = final_clean_trees,
  file = "COMADRE_100_phylo.tre" )

```