Appendix S4 - Extracting Avian Phylogenetic and Trait Data

Nicholas J Clark, David J Harris, Ceridwen Fraser

This appendix describes how to calculate species' pairwise phylogenetic and functional distance matrices. These matrices can be accessed in the `BBS.occurrences` library using `data('Bird.phy.distances')` and `data('Bird.ecol.distances')`.

Load the species occurrence dataset, which lists the binary occurrences of 303 avian species (stored in columns) across sites

```
load("./BBS.Data.cleaned/BBS.2003-2009.filtered.Rdata")
BBS.occurrences <- route.presence.absence
```

Create a vector of avian binomial species names and remove the underscore so that names match the format needed for extracting phylogenies from Birdtree.org

```
sp.names <- gsub("_", " ", colnames(BBS.occurrences))
```

Export the vector as a one-column .csv file to the `Analysis_data` folder created in `Appendix S3`

```
write.csv(sp.names, "./Analysis_data/sp.names.csv",
    row.names = FALSE)
```

Once exported, copy the species names and paste into the `Select species` form on the Phylogeny Subsets page at Birdtree.org. We downloaded 100 trees from the *Ericsson All Species* dataset for our analyses. Once processed, save the resulting .nex file to the `Analysis_data` folder as `sp.trees.nex` and read in the multiphylo object using functions in the `ape` package

```
sp.trees <- ape::read.nexus("sp.trees.nex")
```

Now check to make sure that all of the species names are represented in the tree. Here, we have to include the underscore once again, as this is included in Birdtree.org phylogeny subsets. This call should return `TRUE` if there are no unmatched names

```
sp.names.underscore <- gsub(" ", "_", sp.names)
all(sp.names.underscore %in% sp.trees[[1]]$tip.label)
```

Calculate mean pairwise phylogenetic distances from the distribution of 100 avian trees and scale to range [0, 1]

```
sp.tree.cophenetics <- lapply(seq_along(sp.trees),
    function(x) {
        cophenetics <- ape::cophenetic.phylo(sp.trees[[x]])
        cophenetics <- cophenetics/max(cophenetics)
    })
sp.phy.distances <- Reduce(`+`, sp.tree.cophenetics)/length(sp.tree.cophenetics)
```

Finally, it will be useful to ensure that rows and columns of the phylogenetic distance matrix are ordered to match the column orders in the BBS species occurrence and abundance datasets.

```
reorder_ids <- match(sp.names.underscore, rownames(sp.phy.distances))
sp.phy.distances <- sp.phy.distances[reorder_ids, reorder_ids]
```

Next, we gather functional trait data for each species to create a functional distance matrix. First, we need to download trait data from the EltonTraits database

```
temp <- tempfile()
download.file("https://ndownloader.figshare.com/files/5631081",
    temp)
sp.traits <- read.table(temp, header = TRUE, fill = TRUE,
    quote = "\"", stringsAsFactors = FALSE, sep = "\t")
unlink(temp)
```

The species names are stored in the `Scientific` column and do not have an underscore. Now we filter the dataset to only include species included in our BBS observations

```
library(dplyr)
sp.traits <- sp.traits[which(sp.traits$Scientific %in%
    sp.names), ] %>% dplyr::right_join(data.frame(Scientific = sp.names))
```

Gather the desired trait variables that will be used for creating ecological dendrograms. Here, we choose variables to represent species' proportional use of seven different foraging habitat categories and proportional use of ten diet categories, as these categories together should give a reasonable representation of species' local ecological niches

```
hab.dat <- sp.traits[, c(24:30)]
rownames(hab.dat) <- sp.traits$Scientific
diet.dat <- sp.traits[, c(10:19)]
rownames(diet.dat) <- sp.traits$Scientific
```

Use these diet and foraging trait datasets to calculate a Gower's distance matrix. Because these are proportional variables, we use the `prep.fuzzy` argument to calculate the mixed variable distance coefficient

```
sp.ecol.distances <- prepDendrograms(func.datasets = list(hab.dat,
    diet.dat), prep.types = c("prep.fuzzy", "prep.fuzzy"))
```

Again, we should ensure that all species are in the same order as in the BBS datasets. We then need to add the underscore to the ecological distance row and column names so they match the remaining names

```
reorder_ids <- match(sp.names, rownames(sp.ecol.distances))
sp.ecol.distances <- sp.ecol.distances[reorder_ids,
    reorder_ids]
rownames(sp.ecol.distances) <- colnames(sp.ecol.distances) <- sp.names.underscore
```

Save the phylogenetic and functional trait matrices in the `Analysis_data` folder for downstream analysis. Note, the avian distance matrices can be directly accessed in the `BBS.occurrences` library using `data('Bird.phy.distances')` and `data('Bird.ecol.distances')`.

```
save(sp.ecol.distances, sp.phy.distances, file = "./Analysis_data/
    Bird.Distance.matrices.Rdata")
```