

# Tree join and viz

Luna L Sanchez Reyes

2023-04-27

## The package **ggtree** is on Bioconductor

There are a couple of databases for hosting biological software in R, the most famous one is CRAN, but another very reliable one is called BioConductor.

When we are installing packages from CRAN, we use the function `install.packages()`. To install packages from Bioconductor, we use a function from an R package that is called **BiocManager** with the code `install.packages("BiocManager")`

Once that this package is installed, we can use one of its functions that is called `install()` to install packages from Bioconductor directly: `install("ggtree")` If you get the following error:

```
Error: package or namespace load failed for ‘ggtree’ in loadNamespace(j <- i[[1L]], c(lib.loc, .libPaths()),
  there is no package called ‘aplot’
```

You will need to install another package with `install.packages("aplot")`

Exercise 1 Download this phylogenetic tree of species from the Portal Project Teaching Database, by clicking on the link and saving it to your data-raw folder.

Open the file by clicking on its name on the Files tab of RStudio's Plots pane. It should look like this:

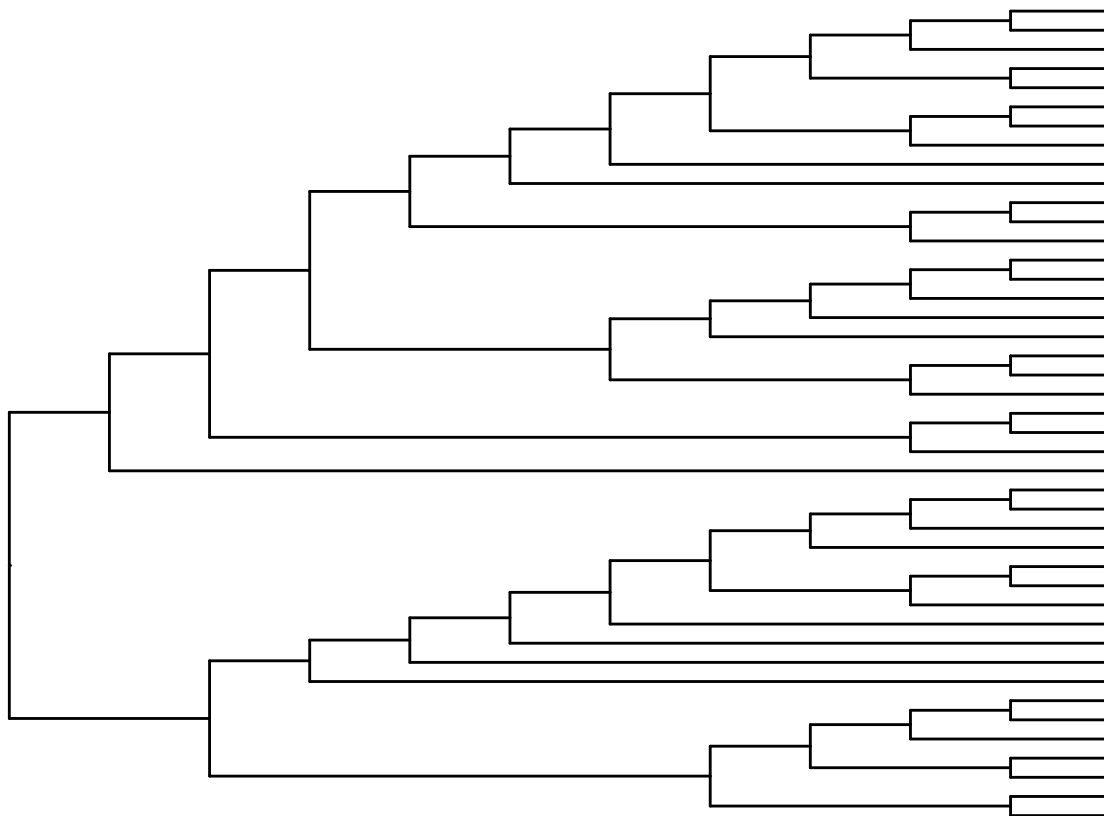
Read the file using a function from the ape package

```
portal_tree <- read.tree(file = "../data-raw/portal-tree.tre")
portal_tree
```

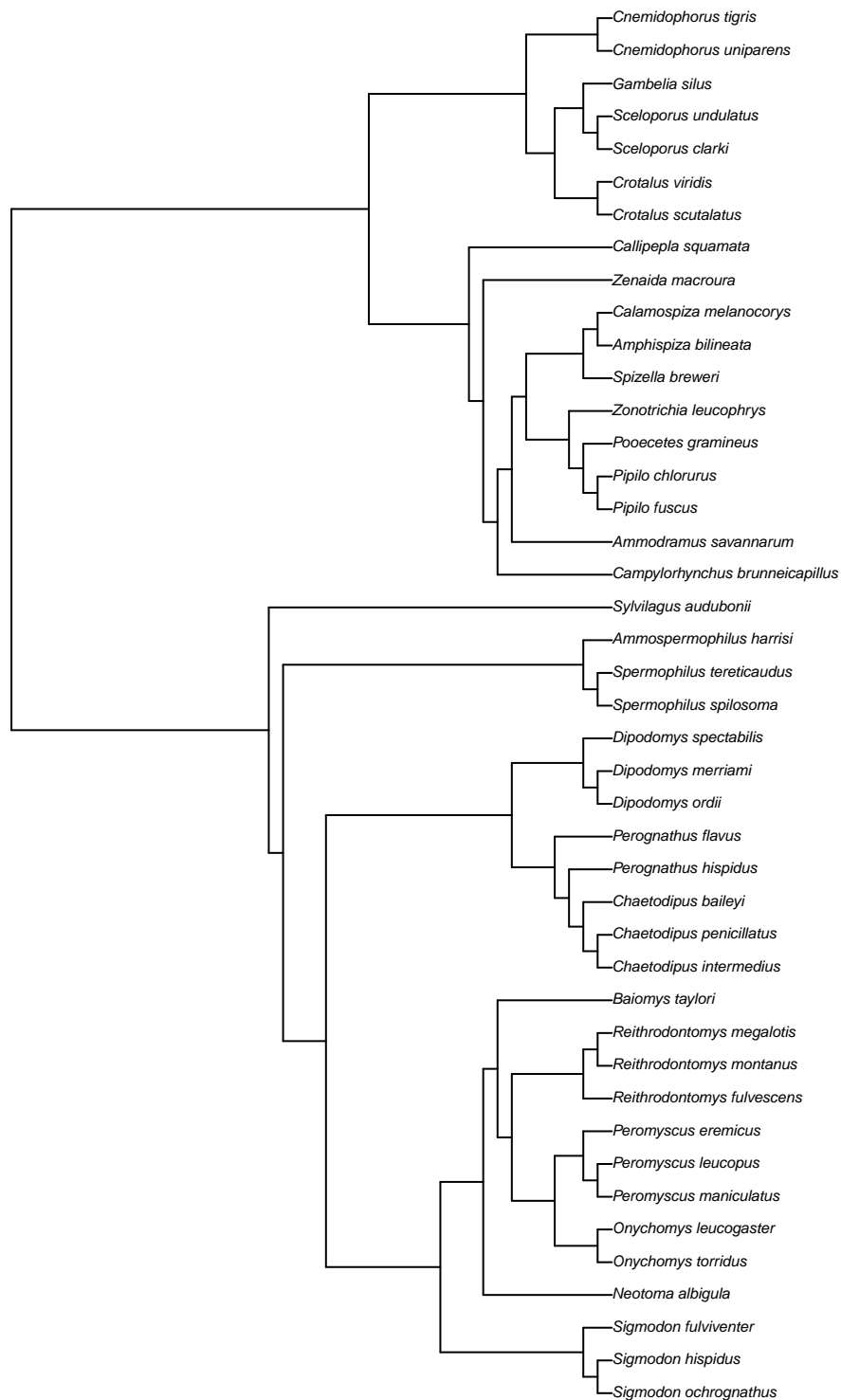
```
##
## Phylogenetic tree with 43 tips and 42 internal nodes.
##
## Tip labels:
##   Sigmodon_ochrognathus, Sigmodon_hispidus, Sigmodon_fulviventer, Neotoma_albigula, Onychomys_torridus
## Node labels:
##   Amniota, mrcaott42ott29157, mrcaott42ott10477, mrcaott42ott48903, mrcaott739ott15637, mrcaott39197
##
## Rooted; no branch lengths.
```

Use the main function of the package **ggtree** (it is also called `ggtree()`) to visualize portal-tree.tre:

```
ggtree(portal_tree)
```



```
plot.phylo(portal_tree, cex = 0.5)
```



## The structure of a tree in R

We have an object that we named `portal_tree`, how do we explore its structure? The `str()` function is a great option

```
str(portal_tree)
```

```
## List of 4
```

```
## $ edge      : int [1:84, 1:2] 44 45 46 47 48 49 50 50 49 48 ...
## $ Nnode     : int 42
## $ node.label: chr [1:42] "Amniota" "mrcaott42ott29157" "mrcaott42ott10477" "mrcaott42ott48903" ...
## $ tip.label : chr [1:43] "Sigmodon_ochrognathus" "Sigmodon_hispidus" "Sigmodon_fulviventer" "Neotoma"
## - attr(*, "class")= chr "phylo"
## - attr(*, "order")= chr "cladewise"
```

What is the class and length of this object:

```
class(portal_tree)
```

```
## [1] "phylo"
```

Homework: How would you define a phylo object class based on its structure?

## Joining a tree and a data table

Download the portal species taxonomy and read it

```
taxonomy <- read.csv("../data-raw/portal-species-taxonomy.csv")
head(taxonomy)
```

```
##               label species_id      genus      species
## 1      Amphispizza_bilineata      AB      Amphispizza      bilineata
## 2      Ammospermophilus_harrisi      AH      Ammospermophilus      harrisi
## 3      Ammodramus_savannarum      AS      Ammodramus      savannarum
## 4      Baiomys_taylori      BA      Baiomys      taylori
## 5      Campylorhynchus_brunneicapillus      CB      Campylorhynchus      brunneicapillus
## 6      Calamospiza_melanocorys      CM      Calamospiza      melanocorys
##      taxa      ott_name approximate_match ott_id is_synonym
## 1      Bird      Amphispizza bilineata      FALSE 292646      FALSE
## 2      Rodent      Ammospermophilus harrisi      TRUE 580350      FALSE
## 3      Bird      Ammodramus savannarum      FALSE 420998      FALSE
## 4      Rodent      Baiomys taylori      FALSE 39171      FALSE
## 5      Bird      Campylorhynchus brunneicapillus      FALSE 800237      FALSE
## 6      Bird      Calamospiza melanocorys      FALSE 948123      FALSE
##      flags
## 1
## 2
## 3
## 4
## 5
## 6
```

What types of functions you would use to joining 2 tables? Functions from the `dplyr` package allow us to do that, for example `inner_join()`

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:ape':
##
##      where
##
## The following objects are masked from 'package:stats':
##
##      filter, lag
```

```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union

surveys <- read.csv(file = "../data-raw/surveys.csv")
species <- read.csv(file = "../data-raw/species.csv")
colnames(surveys)

## [1] "record_id"      "month"          "day"            "year"
## [5] "plot_id"        "species_id"     "sex"            "hindfoot_length"
## [9] "weight"

colnames(species)

## [1] "species_id" "genus"       "species"       "taxa"
# we can use a function that will match the vvalues of tw vectors ot tell us which ones are present in
intersect(colnames(surveys), colnames(species))

## [1] "species_id"

joined <- inner_join(surveys, species, by = "species_id")
head(joined)

##   record_id month day year plot_id species_id sex hindfoot_length weight
## 1         1    7  16 1977      2      NL    M              32      NA
## 2         2    7  16 1977      3      NL    M              33      NA
## 3         3    7  16 1977      2      DM    F              37      NA
## 4         4    7  16 1977      7      DM    M              36      NA
## 5         5    7  16 1977      3      DM    M              35      NA
## 6         6    7  16 1977      1      PF    M              14      NA
##           genus species taxa
## 1   Neotoma albigula Rodent
## 2   Neotoma albigula Rodent
## 3 Dipodomys merriami Rodent
## 4 Dipodomys merriami Rodent
## 5 Dipodomys merriami Rodent
## 6 Perognathus flavus Rodent
```

Apply the same concept to join a phylogeetic tree and a table; the only difference is that we can't do inner\_join, we can only do a left join:

```
left_join(portal_tree, taxonomy, by = "label")

## Error in `dplyr::left_join()`:
## ! ``...` must be empty.
## x Problematic argument:
## * ..1 = !!!dots
## i Did you forget to name an argument?
```