HW1: phylogenetics

Maria Firulyova

08/09/2020

```
library(ape)
library(dplyr)
library(ggplot2)
library(ggtree)
```

1: read the tree in newick format from url to tree1 object

```
set.seed(42)
tree1 <- ape::read.tree('https://www.jasondavies.com/tree-of-life/life.txt')
str(tree1)
```

```
## List of 5
##  $ edge       : int [1:380, 1:2] 192 193 194 195 196 197 198 199 200 201 ...
##  $ edge.length: num [1:380] 1.3496 0.0265 0.0347 0.0555 0.0443 ...
##  $ Nnode      : int 190
##  $ node.label : chr [1:190] "" "Bacteria" "66" "74" ...
##  $ tip.label  : chr [1:191] "Escherichia_coli_EDL933" "Escherichia_coli_O157_H7" "Escherichia_coli_O6" ...
##  - attr(*, "class")= chr "phylo"
##  - attr(*, "order")= chr "cladewise"
```

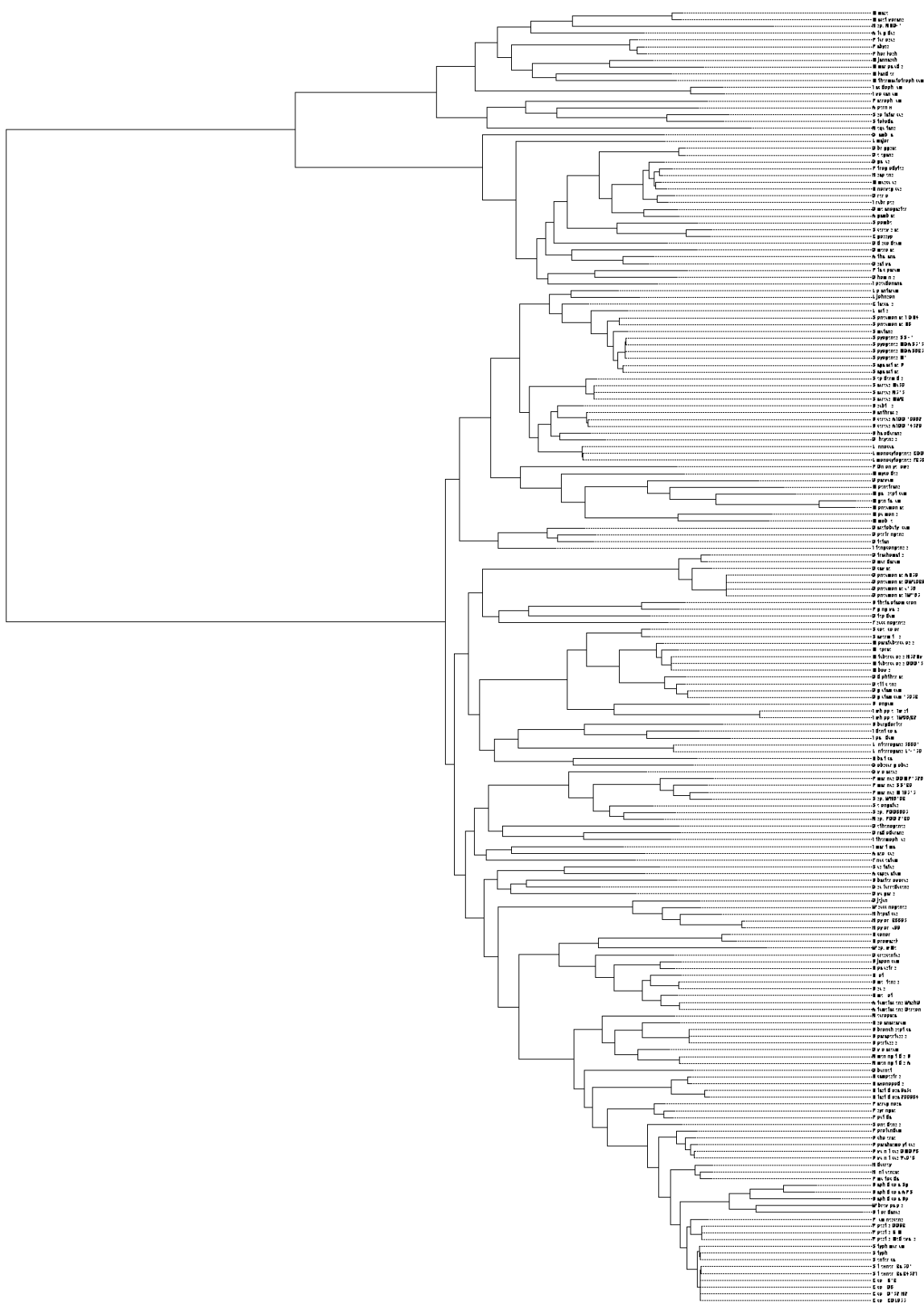Since `tip.labels` are quite large, it can introduce some problems at the visualization step

To handle with this problem, I decide to rename `tip.labels`

```
tree1$tip.label <- stringr::str_replace(tree1$tip.label, '[a-z]*_', '_')
str(tree1)
```

```
## List of 5
##  $ edge       : int [1:380, 1:2] 192 193 194 195 196 197 198 199 200 201 ...
##  $ edge.length: num [1:380] 1.3496 0.0265 0.0347 0.0555 0.0443 ...
##  $ Nnode      : int 190
##  $ node.label : chr [1:190] "" "Bacteria" "66" "74" ...
##  $ tip.label  : chr [1:191] "E_coli_EDL933" "E_coli_O157_H7" "E_coli_O6" "E_coli_K12" ...
##  - attr(*, "class")= chr "phylo"
##  - attr(*, "order")= chr "cladewise"
```

2. Draw the tree using ape functionality

```
plot(tree1, no.margin=T, align.tip.label= T, cex=.6, font=1, use.edge.length = F)
```

3. Save this tree in raster format (png) and vector (svg or pdf)

```r
raster_tree <- function(tree, tree_name, height, width, cex=2) {
  png(tree_name, height = height, width = width)
  plot.phylo(tree, no.margin=T, align.tip.label= T, cex=cex, font=1)
  dev.off()
}

vector_tree <- function(tree, tree_name, height, width, cex=2) {
  svg(tree_name, height = height, width = width)
  plot.phylo(tree, no.margin=T, align.tip.label= T, cex=cex, font=1)
  dev.off()
}

dir.create('plots/tree1', recursive = T)
raster_tree(tree1, "plots/tree1/tree1.png", height=1873.597, width=1302.835, cex=.6)
```

```
## pdf
##   2
```

```r
vector_tree(tree1, "plots/tree1/tree1.svg", height=21, width=14, cex=.6)
```

```
## pdf
##   2
```

```r
knitr::include_graphics("plots/tree1/tree1.svg")
```

4. Draw this tree unrooted or circular using ape functionality

```
plot(unroot(tree1), type="unrooted", no.margin=T, lab4ut="axial", edge.width=2)
```
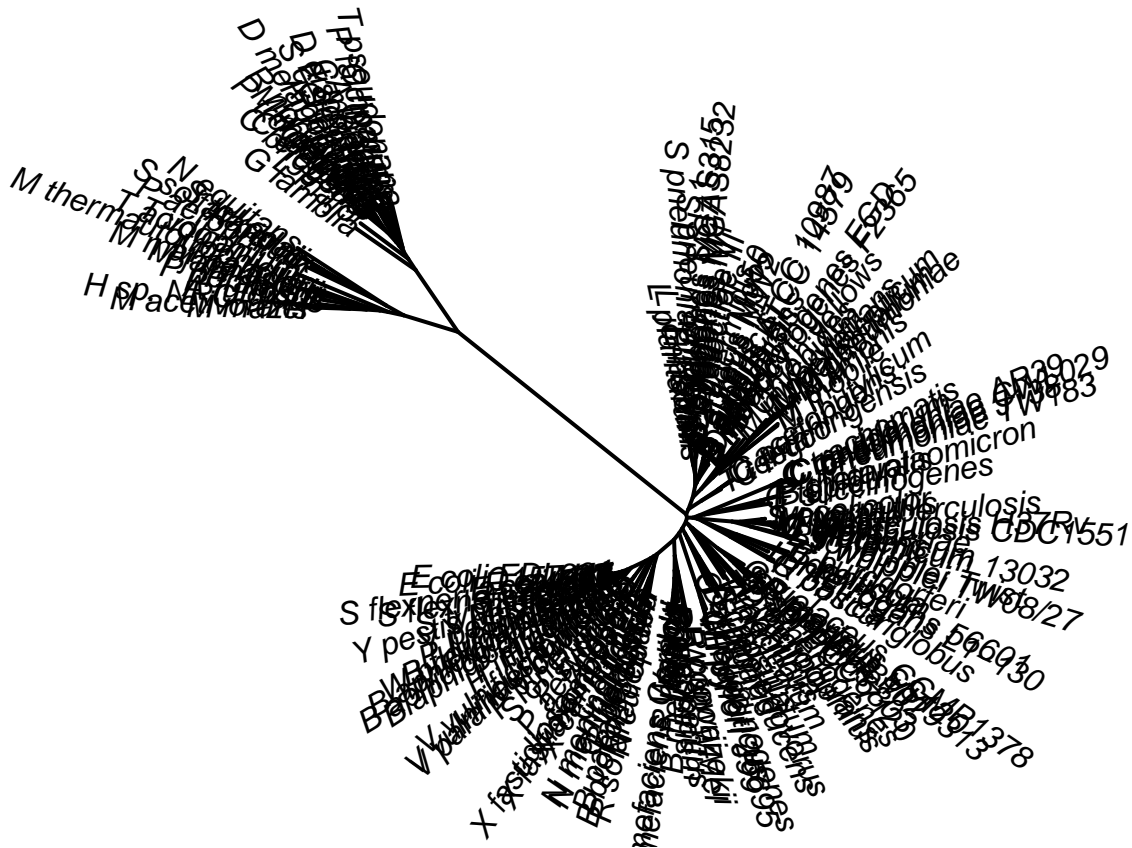

5. Create a new tree2 object

```
tree2 <- read.tree(text="(((A, B), (C, D)), E);")
```

6, 8. Draw the new tree using the standard function from the ape package and save in raster and vector formats (lavels are visible)
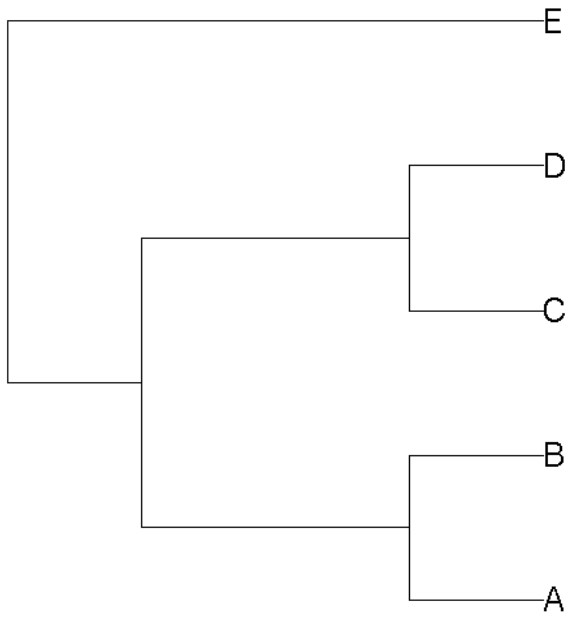
```
dir.create('plots/tree2', recursive = T)
raster_tree(tree2, "plots/tree2/tree2.png", 480, 480)
```

```
## pdf
##   2
```

```
vector_tree(tree2, "plots/tree2/tree2.svg", 4, 4)
```
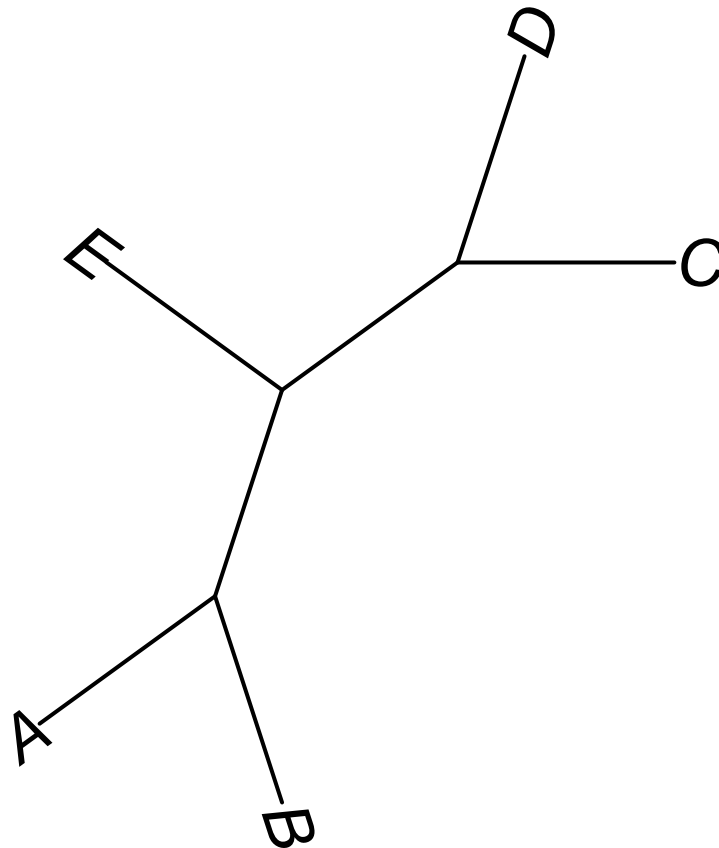
```
## pdf
##   2
```

```
knitr::include_graphics("plots/tree2/tree2.svg")
```
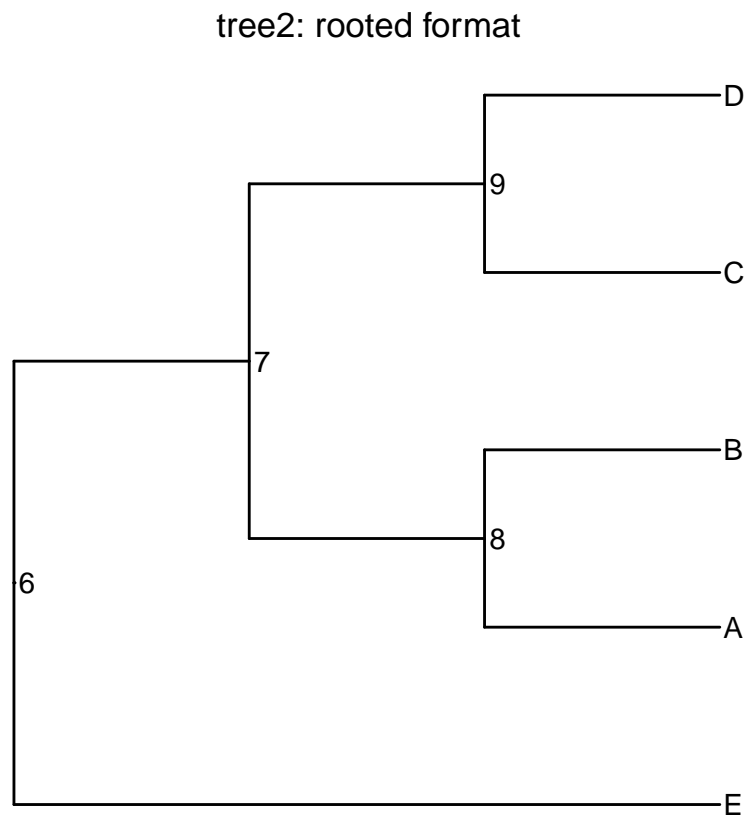
7. Unrooted tree

```
plot(unroot(tree2), type="unrooted", no.margin=T, lab4ut="axial", edge.width=2, cex=2)
```

9. Draw tree object using ggtree package

```
ggtree(tree2)+
  geom_text2(aes(subset=!isTip, label=node), hjust=-.3)+
  geom_tiplab()+
  theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))+
  ggtitle('tree2: rooted format')
```

tree2: rooted format


```
ggtree(tree2, layout = 'circular')+
  geom_text2(aes(subset=!isTip, label=node), hjust=-.3)+
  geom_tiplab()+
  theme(aspect.ratio = 1, plot.title = element_text(hjust = 0.5))+
  ggtitle('tree2: circular format')
```

tree2: circular format

E

A

8

6

D

7

9

B

C