ggtree Plotting Demo

Don

9/28/2017

Contents

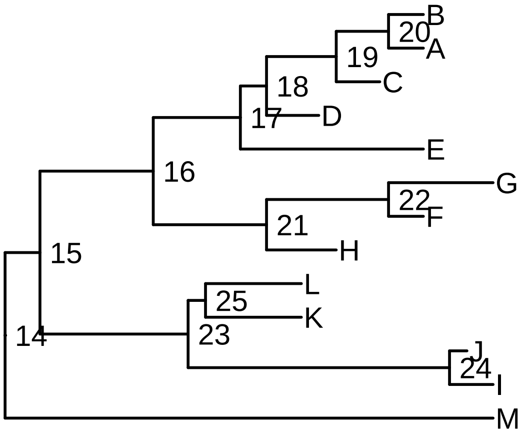
Load a bunch of libraries. Use `install.packages("my_library")` if you haven't got these installed. Protip: `suppressMessages()` hides all the messages that tend to pop up in the console when libraries are loaded.

```
suppressMessages(library("ape"))
suppressMessages(library("ggplot2"))
suppressMessages(library("ggtree"))
```

Basics

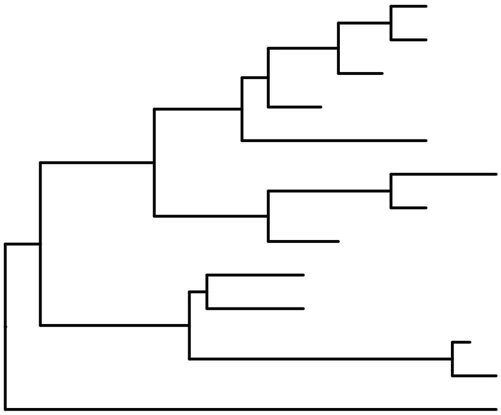
Read in a sample tree, and plot it with node labels. It seems that ggtree automatically enumerates nodes, so that they can be retrieved by their number. Plot the tree first using `ggtree()...`, with the nodes labelled, to get the node number of the node you wish to collapse. Leave out `+ geomtiplab()` if you don't need the tip labels.

```
nwk <- system.file("extdata", "sample.nwk", package="treeio")
tree <- read.tree(nwk)
ggtree(tree) + geom_text2(aes(subset=!isTip, label=node), hjust=-.3) + geom_tiplab()
```



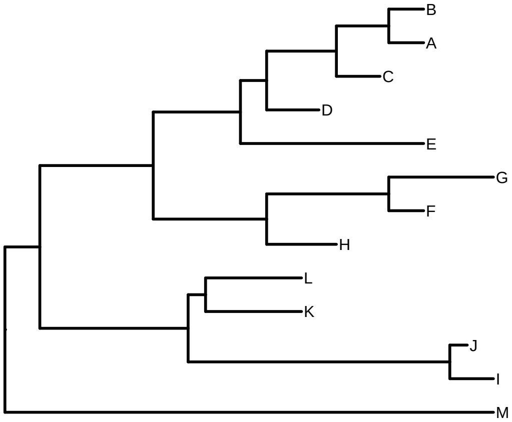
Use the function `ggtree()` to plot. The syntax of `ggtree()` is such that features which you want to attach to your plot are appended with the plus sign `+`, so you'll have something like `ggtree(my_tree) + <new_tree_feature>`. For instance, just calling `ggtree()` will plot only the branches:

```
ggtree(tree)
```



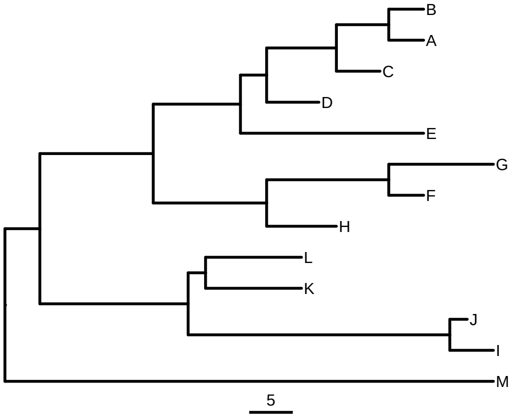
To display tip labels, I add on a `+ geom_tiplab()`, setting the fontsize to 2:

```
ggtree(tree) + geom_tiplab(size=2)
```



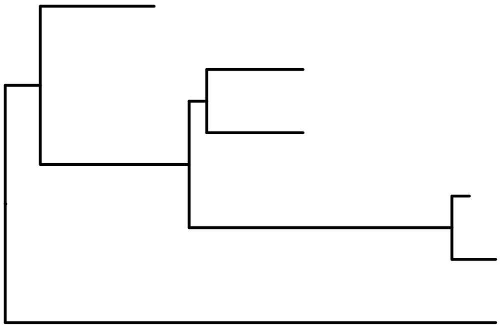
Next, I wish to display the scale bar with `+ geom_treescale()`. I also fiddle with some fontsizes, using `size=` or `fontsize=`, and linewidths with `lwd=`.

```
ggtree(tree, lwd=0.5) + geom_tiplab(size=2) + geom_treescale(fontsize=2)
```



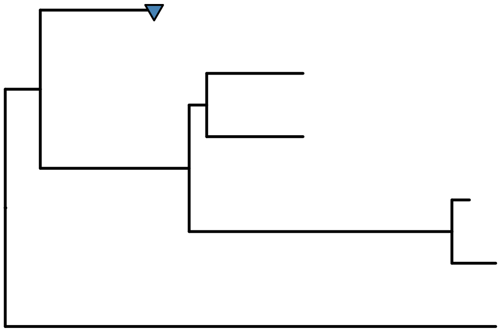
Collapsing a node: identify the node that you wish to collapse by its node number, and apply the `collapse()` function to the `ggtree()` function call, using the `%>%` operator (I have no idea what that is). The following example collapses node 16:

```
ggtree(tree) %>% collapse(node=16)
```



To draw a little shape to represent the collapsed node, use `+ geom_point2`:

```
ggtree(tree) %>% collapse(node=16) +
  geom_point2(aes(subset=(node == 16)), size=2, shape=25, fill="steelblue")
```



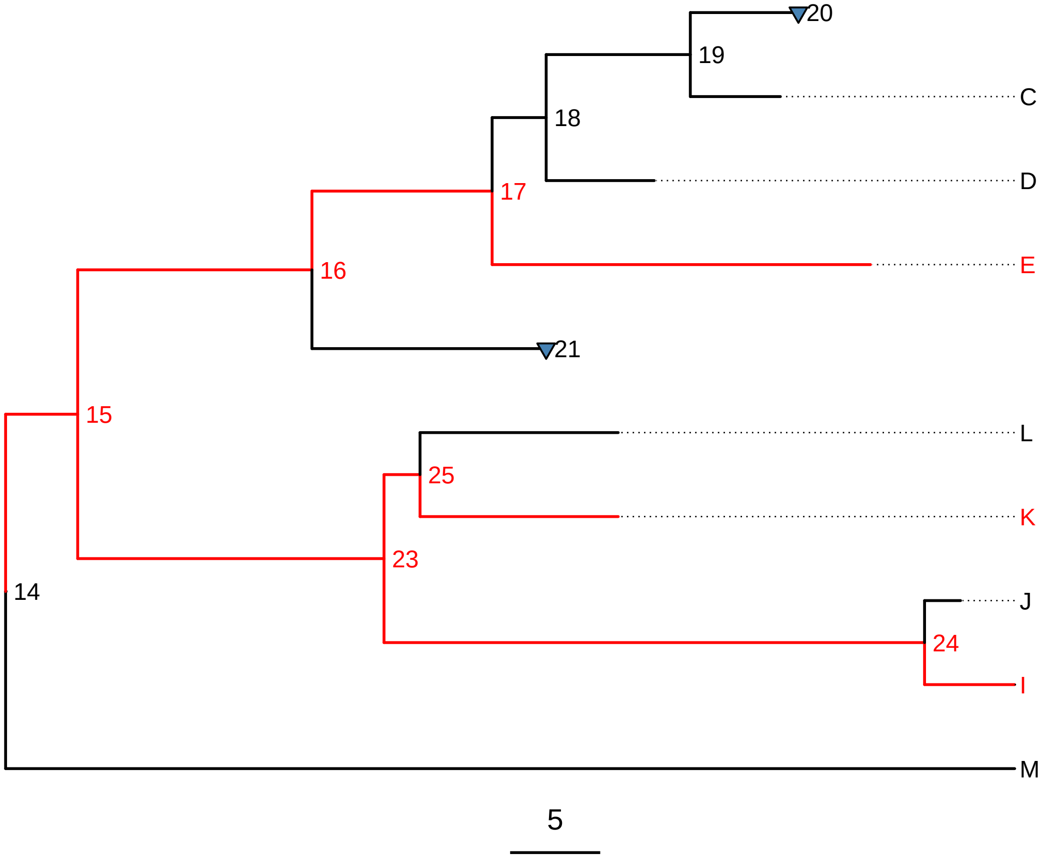
Single Plot Example

Let's do several things: - Highlight tips K, I and G, using `focus()` - Collapse nodes 20 and 21

Group the nodes to highlight into a group, by applying the `groupOTU` function to the tree. This will return some kind of (probably) inconsequential error, complaining about the `size` parameter being called multiple times.

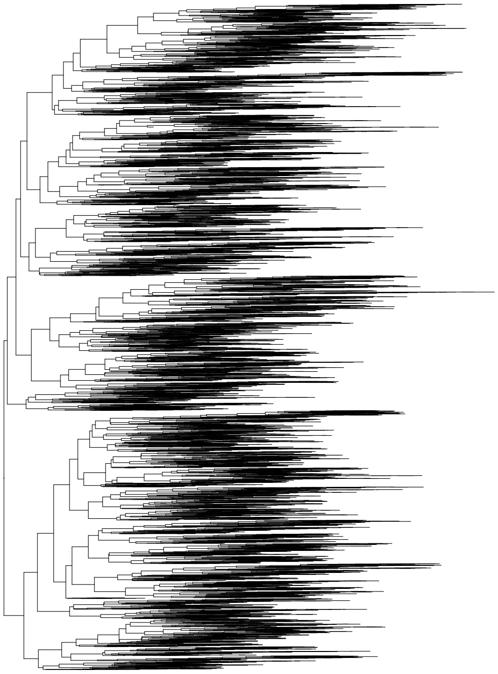
```
tips_to_highlight <- c("K", "I", "E")
tree <- groupOTU(tree, focus=tips_to_highlight)

node_to_collapse1 <- 21
node_to_collapse2 <- 20

# Call ggtree()
ggtree(tree, aes(color=group), ladderize=TRUE) %>% collapse(node=node_to_collapse1) %>% collapse(node=n
  geom_point2(aes(subset=(node == node_to_collapse1)), size=2, shape=25, fill="steelblue") +
  geom_point2(aes(subset=(node == node_to_collapse2)), size=2, shape=25, fill="steelblue") +
  geom_treescale() +
  geom_text2(aes(subset=!isTip, label=node), hjust=-.3, size=3) +
  geom_tiplab(align=TRUE, linesize=0.25, size=3) +
  scale_color_manual(values=c("black", "red"))
```

```
## Warning: The plyr::rename operation has created duplicates for the
## following name(s): (`size`)
```



`ggtree` does pretty well with massive trees, such as this randomly generated tree with 10,000 tips.

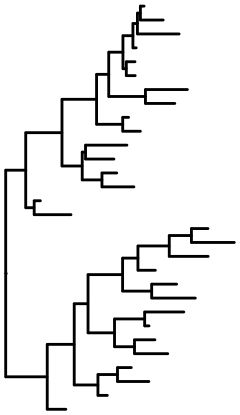
```
ggtree(rtree(10000), lwd=0.1)
```



Multiple Plot Example

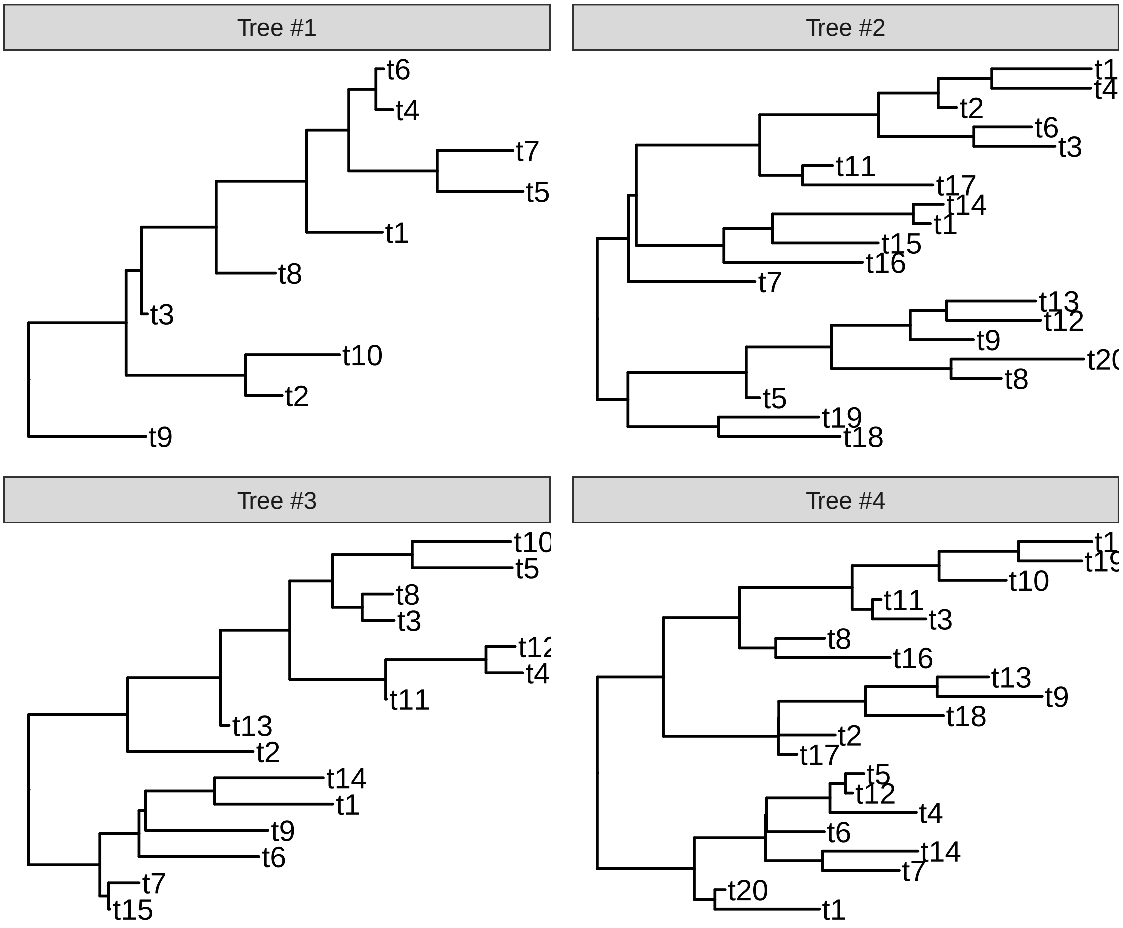
Use the `multiplot()` function:

```
t1 <- ggtree(rtree(30))
t2 <- ggtree(rtree(50))

multiplot(t1, t2, ncol=2)
```



Another example:

```
trees <- lapply(c(10, 20, 15, 20), rtree) # generate 4 random trees
class(trees) <- "multiPhylo" # I think this is coercion
ggtree(trees) + facet_wrap(~.id, scale="free") + geom_tiplab()
```



Note that R seems to only be able to export pdfs to predefined paper proportions (e.g. A3, Letter, A4), so there's an aesthetic limit to the number of trees you can cram into one plot.