

# How to draw a phylogenetic tree

Zuguang Gu <z.gu@dkfz.de>

July 7, 2015

In this short vignette, we will show how to customize the circular style of phylogenetic tree through **circlize** package. Basically, a phylogenetic tree is a dendrogram which is a combination of lines. In R, there are several classes that describe such type of tree such as `hclust`, `dendrogram` and `phylo`. In this example, we will demonstrate how to draw the tree from the `dendrogram` class. Nevertheless, other classes can be converted to `dendrogram` without too much difficulty.

The `bird.orders` data we are using here is from **ape** package. This data set is related to species of birds. We split the tree into six sub trees by `cutree` and convert the data into a `dendrogram` object.

```
library(ape)
data(bird.orders)
hc = as.hclust(bird.orders)
```

```
labels = hc$labels # name of birds
ct = cutree(hc, 6) # cut tree into 6 pieces
n = length(labels) # number of bird species
dend = as.dendrogram(hc)
```

As we mentioned in the main vignette, the x-value for the phylogenetic tree is in fact index. Thus, the x-lim is just the minimum and maximum index of labels in the tree. Since there is only one phylogenetic tree, we only need one “big” sector. In the first track, we plot the name of each bird, with different colors to represent different sub trees.

`max_height` is the maximum height of tree and is used to calculate the height of the track.

```
library(circlize)
circos.par(cell.padding = c(0, 0, 0, 0))
circos.initialize(factors = "a", xlim = c(0, n)) # only one sector
max_height = attr(dend, "height") # maximum height of the trees
circos.trackPlotRegion(ylim = c(0, 1), bg.border = NA, track.height = 0.3,
  panel.fun = function(x, y) {
    for(i in seq_len(n)) {
      circos.text(i-0.5, 0, labels[i], adj = c(0, 0.5),
        facing = "clockwise", niceFacing = TRUE,
        col = ct[labels[i]], cex = 0.7)
    }
  })
```

In the above code, set `xlim` to `c(0, n)` is very important because the leaves of the dendrogram are drawn at `x = seq(0.5, n - 0.5)`.

In the second track, we plot the circular dendrogram by `circos.dendrogram`. You can render the dendrogram by **dendextend** package (figure 1 A).

```
suppressPackageStartupMessages(library(dendextend))
dend = color_branches(dend, k = 6, col = 1:6)
circos.trackPlotRegion(ylim = c(0, max_height), bg.border = NA,
  track.height = 0.4, panel.fun = function(x, y) {
    circos.dendrogram(dend, max_height = max_height)
```

```
})  
circos.clear()
```

By default, dendrograms are facing outside of the circle. You can set facing argument to `inside` to make them facing inside. but of course you need to adjust the name track to the inside of the dendrogram track (figure 1 B).

```
circos.dendrogram(dend, max_height = max_height, facing = "inside")
```

```
circos.par(cell.padding = c(0, 0, 0, 0))  
circos.initialize(factors = "a", xlim = c(0, n)) # only one sector  
circos.trackPlotRegion(ylim = c(0, max_height), bg.border = NA,  
  track.height = 0.4, panel.fun = function(x, y) {  
    circos.dendrogram(dend, max_height = max_height, facing = "inside")  
  })  
circos.trackPlotRegion(ylim = c(0, 1), bg.border = NA, track.height = 0.3,  
  panel.fun = function(x, y) {  
    for(i in seq_len(n)) {  
      circos.text(i-0.5, 1, labels[i], adj = c(1, 0.5),  
        facing = "clockwise", niceFacing = TRUE,  
        col = ct[labels[i]], cex = 0.7)  
    }  
  })  
})  
circos.clear()
```

Figure 1 C is the normal dendrogram.

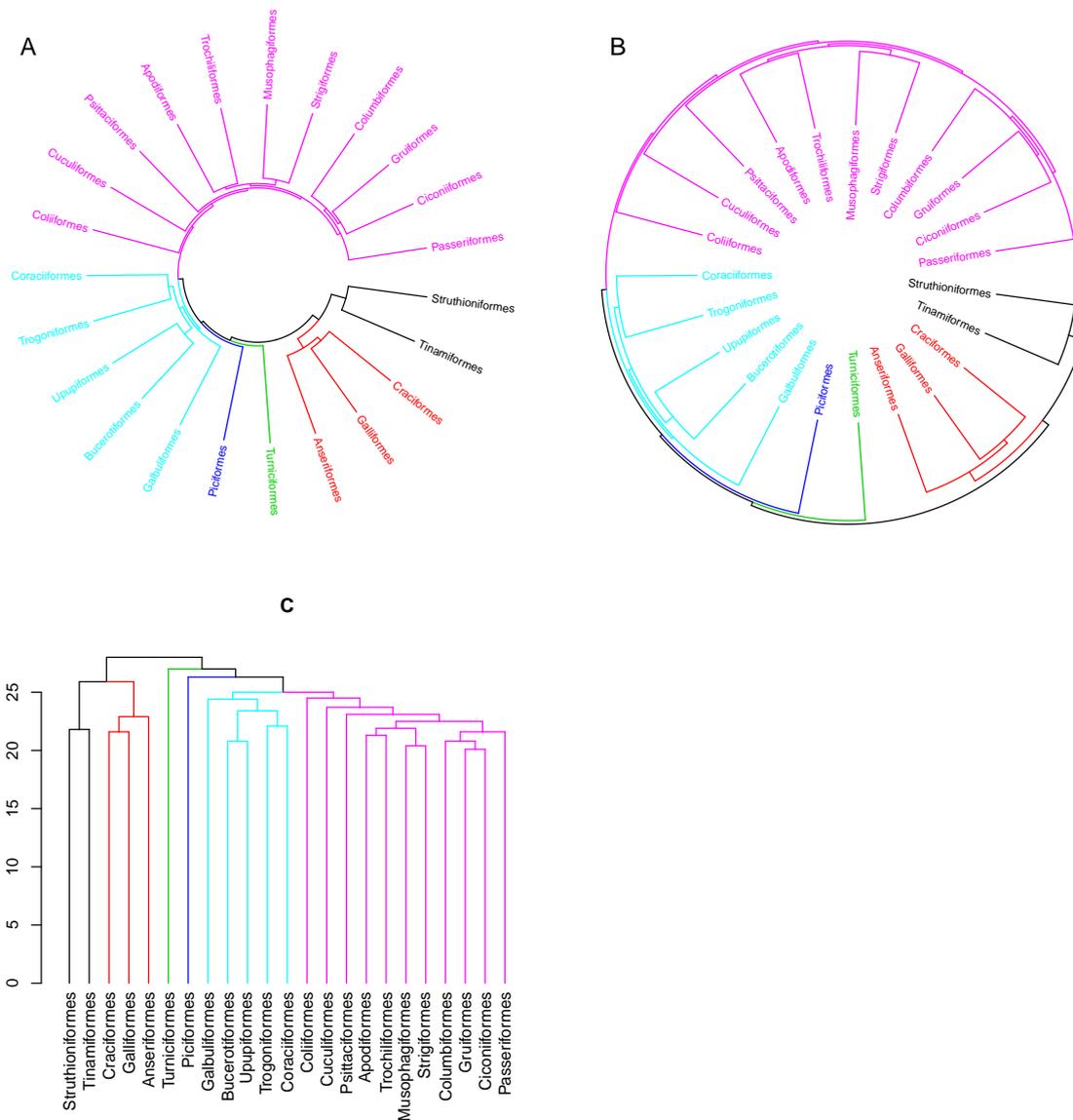


Figure 1: A simple phylogenetic tree. A: circular style; B) dendrogram is facing inside of the circle; C: normal style.