

Tree visualization

Code ▾

Xiaomi Liu

Hide

```
library(ggplot2)
library(ggtree)
```

```
Warning: package 'ggtree' was built under R version 4.3.2
```

Hide

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.3.2
```

Hide

```
library(treeio)
library(RColorBrewer)
```

Hide

```
#read in tree file
tree <- read.tree("./iqtree_output/cov19_tree.treefile")
```

Color by state

Hide

```
#get tip info
tip <- tree$tip.label
#read in state info
info <- read.table("../sample_info2.txt", header = T)
#add reference strain info at the 1st row
info <- rbind(c("coronavirus_Wuh", "WUHAN", "NO"),info)
#build base circular tree
circ <- ggtree(tree,layout = "circular", branch.length = "none", size = 0.01)
```

Warning: `aes_()` was deprecated in ggplot2 3.0.0.

Please use tidy evaluation idioms with `aes()`Warning: Arguments in `...` must be used.

✖ Problematic arguments:

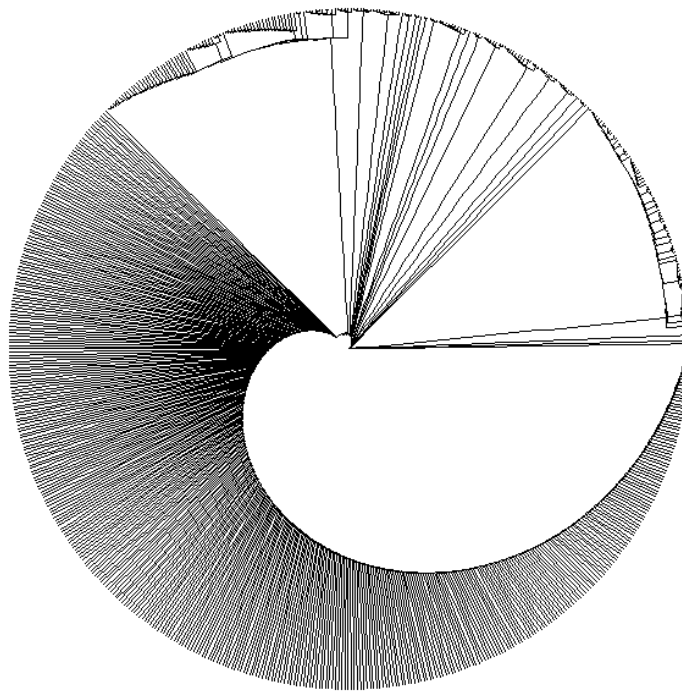
- as.Date = as.Date
- yscale_mapping = yscale_mapping
- hang = hang
- size = 0.01

ⓘ Did you misspell an argument name?Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.

Please use `linewidth` instead.

Hide

circ



Hide

```
#set state at group information
group <- info$state[match(tip, info$id)]
#create data frame to store group information
state <- data.frame(State = group, row.names = tree$tip.label)
#assign color
n <- length(unique(group))
col_pals = brewer.pal.info
col_vector = unlist(mapply(brewer.pal, col_pals$maxcolors, rownames(col_pals)))
colors <- sample(col_vector, n)

#make a tree colored by state
p1 <- gheatmap(circ, state, offset=.8, width=.3, colnames = FALSE) +
  scale_fill_manual(values = colors) +
  labs(fill = "State") +
  guides(fill = guide_legend(override.aes = list(size=3)))
```

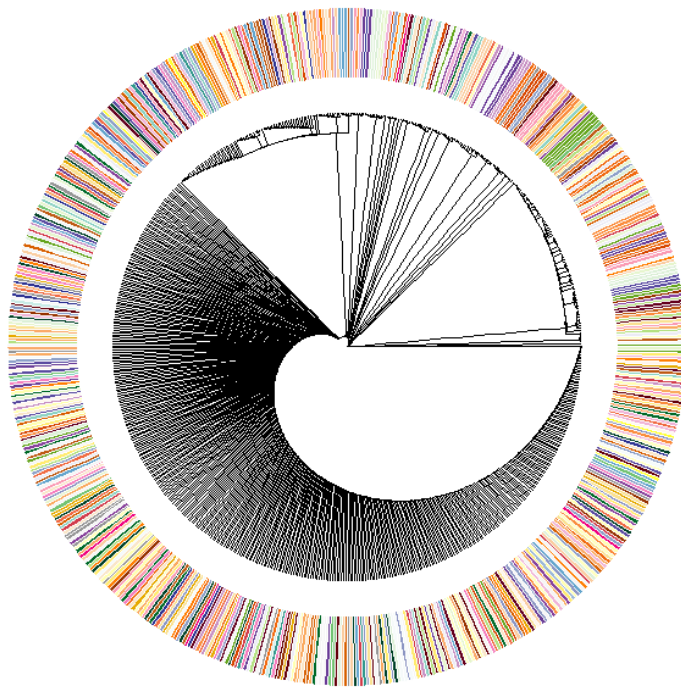
Scale for y is already present.

Adding another scale for y, which will replace the existing scale. Scale for fill is already present.

Adding another scale for fill, which will replace the existing scale.

[Hide](#)

```
p1
#save
ggsave(filename = "./tree_plot/tree_color_by_state.pdf", p1, width = 10, height = 8, units = "in")
```



State

AK	LA	OK
AL	MA	OR
AR	MD	PA
AZ	ME	PR
CA	MI	RI
CO	MN	SC
CT	MO	SD
DE	MS	TN
FL	MT	TX
GA	NC	UT
HI	NE	VA
IA	NH	VT
ID	NJ	WA
IL	NM	WI
IN	NV	WV
KS	NY	WY
KY	OH	

Color by region

[Hide](#)

```
#get map data for USA
library(maps)
us_states <- map_data("state")
unique(us_states$region)
```

[1] "alabama"	"arizona"	"arkansas"	"california"
"colorado"			
[6] "connecticut"	"delaware"	"district of columbia"	"florida"
"georgia"			
[11] "idaho"	"illinois"	"indiana"	"iowa"
"kansas"			
[16] "kentucky"	"louisiana"	"maine"	"maryland"
"massachusetts"			
[21] "michigan"	"minnesota"	"mississippi"	"missouri"
"montana"			
[26] "nebraska"	"nevada"	"new hampshire"	"new jersey"
"new mexico"			
[31] "new york"	"north carolina"	"north dakota"	"ohio"
"oklahoma"			
[36] "oregon"	"pennsylvania"	"rhode island"	"south carolina"
"south dakota"			
[41] "tennessee"	"texas"	"utah"	"vermont"
"virginia"			
[46] "washington"	"west virginia"	"wisconsin"	"wyoming"

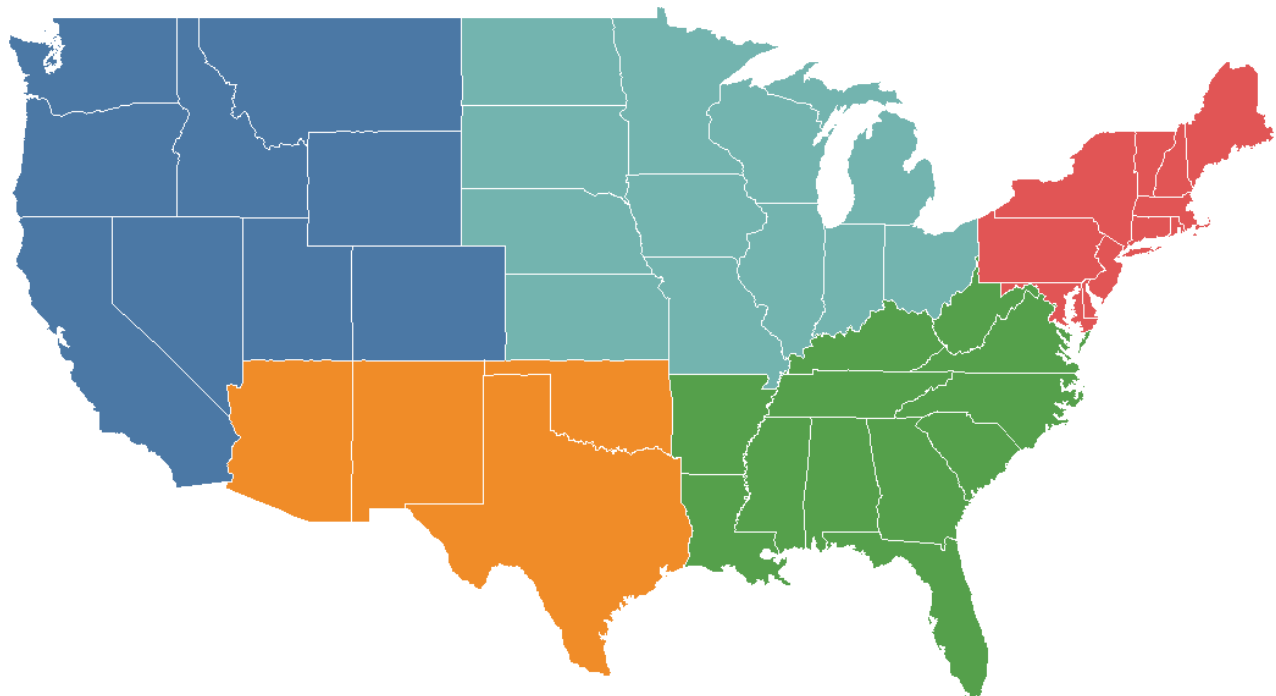
Hide

```
#divide in subregion
west <- c("washington","oregon", "california","idaho" ,"montana", "wyoming", "utah", "nevada",
"colorado")
southwest <- c("arizona", "new mexico", "oklahoma", "texas")
midwest <- c("north dakota", "south dakota", "kansas", "kansas", "minnesota", "iowa", "missouri",
"wisconsin", "illinois", "michigan", "indiana", "ohio", "nebraska")
northeast <- c("maryland", "delaware", "new jersey", "connecticut", "rhode island", "new hampshire",
"massachusetts", "vermont", "maine", "new york", "pennsylvania")
for (i in 1: nrow(us_states)){
  if (us_states$region[i] %in% west){
    us_states$subregion[i] <- "W"
  }
  else if(us_states$region[i] %in% southwest){
    us_states$subregion[i] <- "SW"
  }
  else if(us_states$region[i] %in% midwest){
    us_states$subregion[i] <- "MW"
  }
  else if(us_states$region[i] %in% northeast){
    us_states$subregion[i] <- "NE"
  }
  else{
    us_states$subregion[i] <- "SE"
  }
}
us_states[us_states$region == "alaska",]
```

0 rows

Hide

```
#make a colored map
colors_named <- c(
  W      = "#4E79A7",
  SW     = "#F28E2B",
  MW     = "#76B7B2",
  NE     = "#E15759",
  SE     = "#59A14F",
  WUHAN  = "#B07AA1",
  Other  = "#9C755F")
us_states$subregion <- factor(us_states$subregion, levels = c("W", "SW", "MW", "NE", "SE"))
map <- ggplot(data = us_states, aes(x = long, y = lat, group = group, fill = subregion))+
  geom_polygon(color = "white", size = 0.1, show.legend = F) +
  scale_fill_manual(values = colors_named[1:5]) +
  coord_fixed(1.3) +
  theme_bw() +
  theme(panel.background = element_blank(), panel.border = element_blank(), panel.grid = element
_blank(), axis.title = element_blank(), axis.text = element_blank(), axis.ticks = element_blank
())
map
ggsave(filename = "./tree_plot/usa_color_by_reagion.pdf", map, width = 4, height = 3)
```



Hide

```
#make a dataframe to store color group information again
ring2 <- state
ring2$region <- NA
region <- unique(us_states[,5:6])
state_abb <- read.table("./state_abbrev.txt", header = F, sep = "\t")
for(i in 2:nrow(ring2)) {
  abb <- ring2$State[i] %>% as.character()
  full <- state_abb[state_abb$V2 == abb, 1] %>% as.character() %>% tolower()
  if(full == "alaska"){
    reg <- "Other"
  }
  else if(full == "puerto rico"){
    reg <- "Other"
  }
  else if(full == "hawaii"){
    reg <- "Other"
  }
  else{
    reg <- region[region$region == full, 2] %>% as.character()
  }
  ring2$region[i] <- reg
}
ring2$region[1] <- "WUHAN"
ring2 <- dplyr::select(ring2, region)
ring2$region <- factor(ring2$region, levels = c("W", "SW", "MW", "NE", "SE", "WUHAN", "Other"))
#make a tree colored by sub region
library(ggnewscale)
p2 <- gheatmap(circ, ring2, offset=.8, width=.3, colnames = FALSE) +
  scale_fill_manual(values = colors_named, breaks = c("W", "SW", "MW", "NE", "SE", "WUHAN", "Other")) +
  labs(fill = "Region") +
  guides(fill = guide_legend(override.aes = list(size=3)))
```

Scale for y is already present.

Adding another scale for y, which will replace the existing scale. Scale for fill is already present.

Adding another scale for fill, which will replace the existing scale.

Hide

```
p2
#save
ggsave(filename = "./tree_plot/tree_color_by_region.pdf", p2, width = 10, height = 8, units = "in")
```

