

Phylogenetic placement visualization with treeio-ggtree method

0.1 Favorable placement filtration after parsing jplace file

```

# Required packages
library(treeio)
library(ggtree)
library(jsonlite)
library(dplyr)
library(ggtreeExtra)
library(RColorBrewer)
library(colorspace)
library(ggplot2)
library(aplot)

# Read jplace file
jtree <- jsonlite::fromJSON("../exampledata/Holomycota/HolomycotaV4_alignedtrim.jplace")

# Convert jplace to phylo object
phylo <- treeio::jplace_treetext_to_phylo(jtree$tree)

# Extract placement information from jplace object
placements <- treeio::extract.placement(jtree, phylo)

# Filter placements, selecting the maximum like_weight_ratio for each sequence
filtered_placements <- dplyr::group_by(placements, .data$name) %>%
  filter(.data$like_weight_ratio == max(.data$like_weight_ratio))

# Create a data frame to store the group and corresponding likelihood weight ratios
dat <- data.frame(
  group = c(rep("Before", length(placements$like_weight_ratio)), rep("Filtered", length(filtered_placements$like_weight_ratio))),
  likelihood_weight_ratio = c(placements$like_weight_ratio, filtered_placements$like_weight_ratio)
)

# Plot the distribution of likelihood weight ratios using a histogram with ggplot2
p1 <- ggplot(data = dat, aes(x = likelihood_weight_ratio)) +
  geom_histogram(data = dat, aes(x = likelihood_weight_ratio, fill = group, ..count..), position = position_dodge()) +
  scale_fill_manual(values = c("#f08c8c", "#8dc5fe")) +
  labs(y = "Number of Placements", x = "Likelihood Weight Ratio") +
  scale_x_continuous(breaks = c(seq(0, 1, 0.1))) +
  theme_bw()

# Convert phylo object to treedata object
tree1 <- treeio::as.treedata(phylo)

# Convert treedata to tibble
td <- dplyr::as_tibble(tree1)

# Calculate the number of placements per node and merge with original data
tree1@data <- dplyr::group_by(filtered_placements, .data$node) %>%
  dplyr::summarize(nplace = n()) %>%
  dplyr::full_join(td, by = 'node') %>%
  dplyr::mutate(nplace = ifelse(is.na(.data$nplace), 0, .data$nplace))

# Use ggtree to plot a circular tree, adjusting node color and size based on the number of placements
p2 <- ggtree::ggtree(tree1, layout = "circular", branch.length = "none", aes(color = nplace, size = nplace)) +
  colorspace::scale_color_continuous_sequential(palette = "Sunset", limits = c(1, 55)) +
  scale_size_continuous(range = c(0.1, 2), limits = c(0, 55),
    guide = guide_legend(reverse = TRUE)) +

```

```

theme(
  legend.position = "right",
  legend.box = "horizontal",
  legend.box.just = "left"
)

# Read group information from CSV file
V4_group <- read.csv(file = "../exampledata/Holomycota/V4_group.csv")

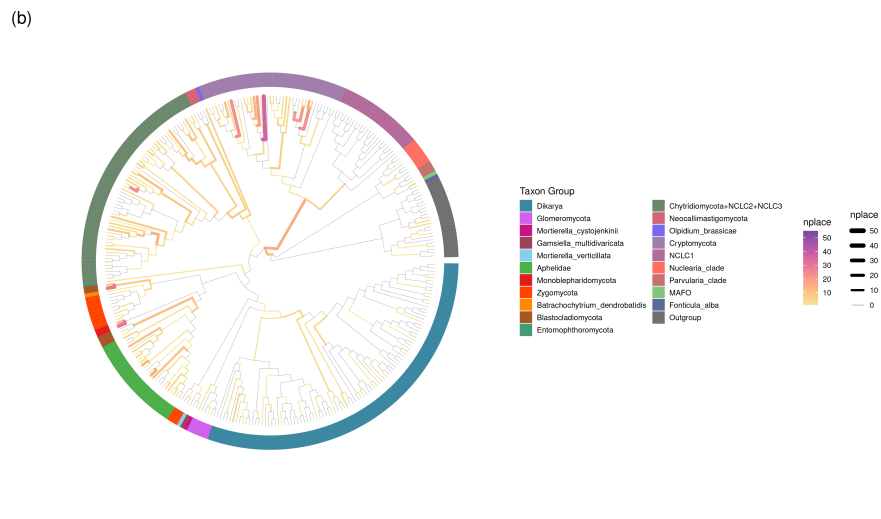
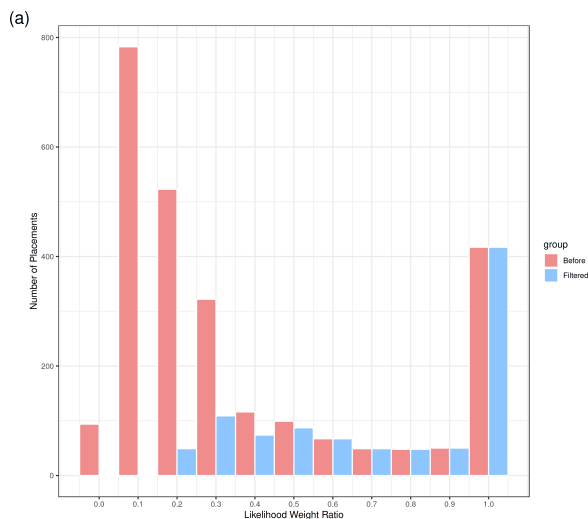
cols <- c(
  "#3B87A2", "#D15FEE", "#C71585", "#9E425A", "#87CEEB",
  "#4DAF4A", "#E41A1C", "#FF4500", "#FF8C00", "#A65628",
  "#449B75", "#6B886D", "#D36477", "#7B68EE", "#A07EAB",
  "#B46B9B", "#FF6F61", "#C67171", "#81c581", "#596A98",
  "#707070"
)

# Get unique levels of the group and reverse the order
levels <- rev(unique(V4_group$group))

# Use ggtree and ggtreeExtra to draw the plot with group information
p3 <- p2 %<+% V4_group +
  ggtreeExtra::geom_fruit(
    geom = geom_tile,
    mapping = aes(fill = factor(group, levels = levels)),
    width = 2,
    offset = 0.1
  ) +
  scale_fill_manual(
    name = "Taxon Group",
    values = cols,
    guide = guide_legend(
      keywidth = 1,
      keyheight = 1,
      order = 3
    ),
    na.translate = FALSE
  )

# Combine p1 and p3 into a single plot and add labels
aplot::plot_list(p1, p3, tag_levels = list(c('(a)', '(b)', '(c)', '(d)'), '1'),
  tag_size = 20)

```



0.2 Utility to explore the placement uncertainty

```
# Required packages
library(ggplot2)
library(treeio)
library(ggtree)
library(jsonlite)
library(dplyr)
library(colospace)
library(ggtreeExtra)
library(aplot)
library(patchwork)

# Read jplace file containing tree data
jtree <- jsonlite::fromJSON("../exampledata/subtree/ppplacer_Amt_subtree.jplace")

# Convert jplace to phylo object
phylo <- treeio::jplace_treetext_to_phylo(jtree$tree)

# Extract placement information from the jplace object
placements <- treeio::extract_placement(jtree, phylo)

# Specify the sequence name of interest
seq_name <- "saltern1"

# Filter placements to keep only those matching the specified sequence name
sgplacement <- dplyr::group_by(placements, .data$name) %>%
  dplyr::filter(.data$name == seq_name)

# Convert phylo object to treedata format
tree1 <- treeio::as_treedata(phylo)
tree1@phylo$tip.label <- gsub("Ostreococcus", "Os", tree1@phylo$tip.label)
# Join filtered placements with the tree data by node
tree2 <- dplyr::left_join(tree1, sgplacement, by = "node")

# Plot a circular tree using ggtree, colored by likelihood weight ratio
p1 <- ggtree::ggtree(tree2, layout = "circular", aes(color = like_weight_ratio)) +
  colospace::scale_color_continuous_sequential(palette = "Sunset", limits = c(0, 1)) +
  ggtitle(seq_name)

# Read taxon label data from CSV file
Amt_tiplabel <- read.csv(file = "../exampledata/subtree/Amt_tiplabel.csv", header = TRUE)

# Display the first few rows of the taxon label data
head(Amt_tiplabel)
```

##	node	tiplabel	Group
## 1	1	Asterochloris Trebouxioephyceae	
## 2	2	Asterochloris Trebouxioephyceae	
## 3	3	Coccomyxa Trebouxioephyceae	
## 4	4	Stichococcus Trebouxioephyceae	
## 5	5	Galdieria	Cyanidiales
## 6	6	Galdieria	Cyanidiales

```
# Define the taxon groups for categorization
taxon_groups <- c("Trebouxioephyceae", "Cyanidiales", "Non-Cyanidiales",
  "Chlamydomonadales", "Tetraselmis ssp.", "Pyramimonas ssp.",
  "Nephroselmis pyriformis CCMP717", "Bathycoccus ssp.",
  "Ostreococcus", "Mantoniella ssp.", "Micromonas ssp.",
  "Embryophyta")
```

```

# Convert Group column to a factor with specified levels
Amt_tiplabel$Group <- factor(Amt_tiplabel$Group, levels = taxon_groups)

# Define color palette for the groups
cols <- c("#FFC125", "#87CEFA", "#7B68EE", "#191970", "#800080",
          "#9ACD32", "#D15FEE", "#FFC0CB", "#EE6A50", "#8DEEEE",
          "#006400", "#800000", "#B0171F")

# Add group information to the tree plot p1
p2 <- p1 %<+% Amt_tiplabel +
  ggtreeExtra::geom_fruit(
    geom = geom_tile,
    mapping = aes(fill = Group),
    width = 0.2,
    offset = 0.01
  ) +
  scale_fill_manual(
    name = "",
    values = cols,
    guide = guide_legend(
      keywidth = 1,
      keyheight = 1,
      order = 3
    ),
    na.translate = FALSE
  )

# Note: viewClade is not applicable for circular trees
# Create a subset of the tree for a specific node
tree3 <- tidytree::tree_subset(tree2, node = 657, levels_back = 0)

# Replace "Ostreococcus" with "Os"
tree3@phylo$tip.label <- gsub("Ostreococcus", "Os", tree3@phylo$tip.label)

# Set options to avoid scientific notation in plot labels
options(scipen = 200)

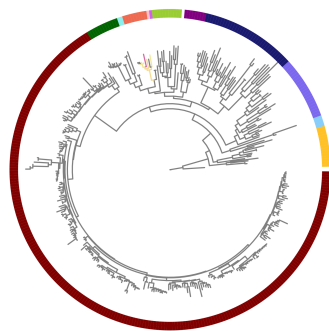
# Round the likelihood weight ratio values and convert them to character for labeling
tree3@extraInfo[, 9] <- tree3@extraInfo[, 9] |>
  as.vector() |>
  unlist() |>
  as.numeric() |>
  round(digits = 5)

# Plot the subsetted tree using a rectangular layout, colored by posterior probabilities
p3 <- ggtree::ggtree(tree3, layout = "rect", aes(color = post_prob), size = 1) +
  colorspace::scale_color_continuous_sequential(palette = "Sunset", limits = c(0, 1)) +
  geom_tiplab(size = 3, color = "black", align=T) +
  xlim(0, 2) +
  geom_tiplab(size = 3, color = "black", align = T) +
  xlim(0, 2) +
  geom_label2(aes(x = branch, label = post_prob), color = 'black', fill = "#dcb1e2af", size = 2)

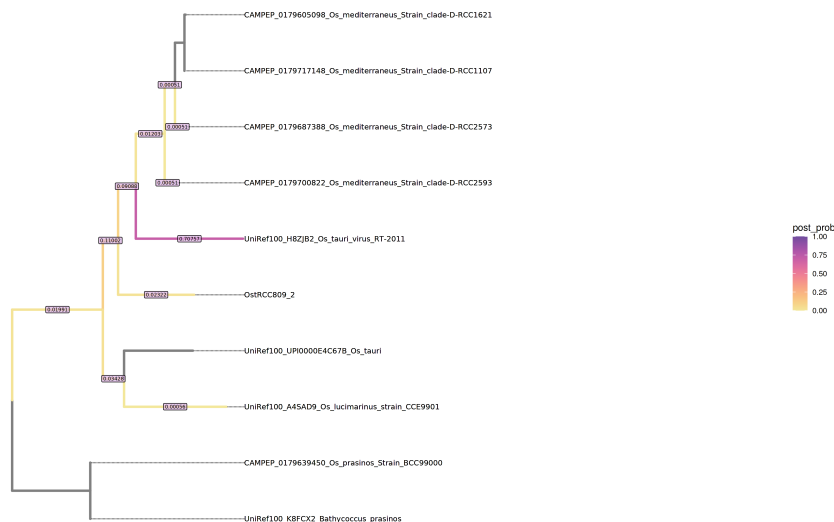
aplot::plot_list(p2, p3,
  tag_levels = list(c('(a)', '(b)'), design = "ABB\nABB")

```

(a)
saltern1



(b)



0.3 Merge placement information to the placement tree

Required packages

```
library(treeio)
library(ggtree)
library(ggplot2)
library(ggtreeExtra)
library(colorspace)
library(dplyr)
library(patchwork)

# Read jplace file containing placement data for the phylogenetic tree
tree1 <- treeio::read.jplace(file = "../exampledata/Mitsi/rsbl20190182supp2.jplace")
```

```
# Read Newick formatted tree file and convert to treedata format
tree2 <- ggtree::read.tree(file = "../exampledata/Mitsi/rsbl20190182supp7.tre")
tree2 <- treeio::as.treedata(tree2)
```

```
# Create a circular tree plot with color and linewidth based on number of placements
p1 <- ggtree::ggtree(tree1, layout = "circular", aes(color = nplace, size = nplace)) +
  scale_color_viridis_c(option = "viridis", limit = c(0.1, 120)) +
  scale_size_continuous(range = c(0, 2))
```

```
# Extract placement information from the jplace tree
pla <- treeio::get.placements(tree1)
```

```
pla$label <- pla$name
pla2 <- dplyr::select(pla, c("name", "label", "likelihood", "like_weight_ratio"))
```

```
# Convert tree2 to a tibble
tree3 <- dplyr::as_tibble(tree2)
```

```
# Merge the placement data (pla2) onto the tree data (tree3) using labels
tree4 <- dplyr::left_join(tree3, pla2, by = "label")
```

```
# Clean up group names by removing suffixes
tree4$group <- gsub("_1", "", tree4$label)
tree4$group <- gsub("_2", "", tree4$group)
```

```
# Extract the second element from the split group names
tree4$group <- lapply(tree4$group, function(x) { unlist(strsplit(x, "_"))[c(2)] })
```

```

# Rename specific groups for clarity
tree4$group <- gsub("Rhabditophora", "Bothrioplana_semperi", tree4$group)

# Define standard groups for categorization
s_group <- c("Catenulida", "Polycladida", "Macrostomorpha", "Prorhynchidae",
            "Proseriata", "Rhabdocoela", "Bothrioplana_semperi", "Fecampiidae",
            "Tricladida", "Prolecithophora", "Monogenea", "Trematoda", "Cestoda")

# Extract OTU groups from the data
otu_group <- unlist(list(tree4[grep("^OTU.", tree4$group), "group"]))
all_group <- c(s_group, otu_group)

# Assign "Outgroup" to other undefined groups
tree4[!(tree4$group %in% all_group), "group"] <- "Outgroup"

# Assign specific group names to certain row indices
tree4[seq(193, 201), "group"] <- "Prorhynchidae"
tree4[seq(249, 252), "group"] <- "Fecampiidae"
tree4[seq(169, 171), "group"] <- "Gnosonesimidae"
tree4[seq(119, 121), "group"] <- "Clade 1"
tree4[seq(174, 192), "group"] <- "Clade 2"

s_group2 <- c("Trematoda", "Cestoda", "Monogenea",
            "Prolecithophora", "Tricladida", "Fecampiidae",
            "Bothrioplana_semperi", "Rhabdocoela", "Proseriata", "Prorhynchidae",
            "Clade 2", "Gnosonesimidae", "Macrostomorpha", "Polycladida",
            "Clade 1", "Catenulida", "Outgroup")

# Set groups that are NA to be NA
tree4[is.na(tree4$label), "group"] <- NA

# Convert the group column to a factor with specified levels
tree4$group <- factor(tree4$group, levels = s_group2)

# Assign a default value to NA likelihood weight ratios
tree4[is.na(tree4$like_weight_ratio), "like_weight_ratio"] <- 0

# Convert the modified tibble back to treedata format
tree5 <- treeio::as.treedata(tree4)

# Create a circular tree plot based on tree5 with specified aesthetics
p2 <- ggtree::ggtree(tree5, layout = "circular") +
  ggtree::geom_tree(
    aes(
      color = like_weight_ratio, # Color branches based on likelihood weight ratio
      size = like_weight_ratio # Size branches based on likelihood weight ratio
    )
  ) +
  colorspace::scale_color_continuous_sequential(palette = "Sunset", limits = c(0.000001, 0.2)) + # Set color
  scale_size_continuous(range = c(0, 1.5), guide = guide_legend(reverse = TRUE)) + # Scale size of branches
  ggtree::geom_tiplab2(data = td_filter(like_weight_ratio > 0), align = TRUE, size = 3, offset = 0.1) # Label

# Scale the clade for specific nodes
p3 <- ggtree::scaleClade(p2,
  node = c(MRCA(tree5, "SA_OTU14871", "SA_OTU158753")),
  scale = 5)

# Scale another clade for specific nodes

```

```
p4 <- ggtree::scaleClade(p3,
  node = c(MRCA(tree5, "SA_OTU22294", "SA_OTU9482")),
  scale = 5)

# Add a out-circle layer to the plot to represent group information with colored tiles
p5 <- p4 +
  ggtreeExtra::geom_fruit(geom = geom_tile,
    mapping = aes(fill = group),
    width = 0.1,
    offset = 0.01)

# Display the final plot
p5
```

