R script for Comparative phylogenomic analyses of SNP versus full locus datasets: insights and recommendations for researchers

Prepared by Jacob S. Suissa

2023-08-30

Overview

This is an R Markdown document for the Cornell SIPS Plant Biology Phylogenetic Journal Club project exploring how SNP vs. locus data and missing data affect phylogeny topology, branch length, nodal support, node dating, and downstream phylogenetic comparative methods.

Load libraries

First load the libraries

```
library(tidyverse)
library(ape)
library(phytools)
library(geiger)
library(ggdist)
library(ggpubr)
library(ggsci)
library(treeio)
library(RColorBrewer)
```

Import data

Read in all the trees and make sure to subset the locus and the SNPS files

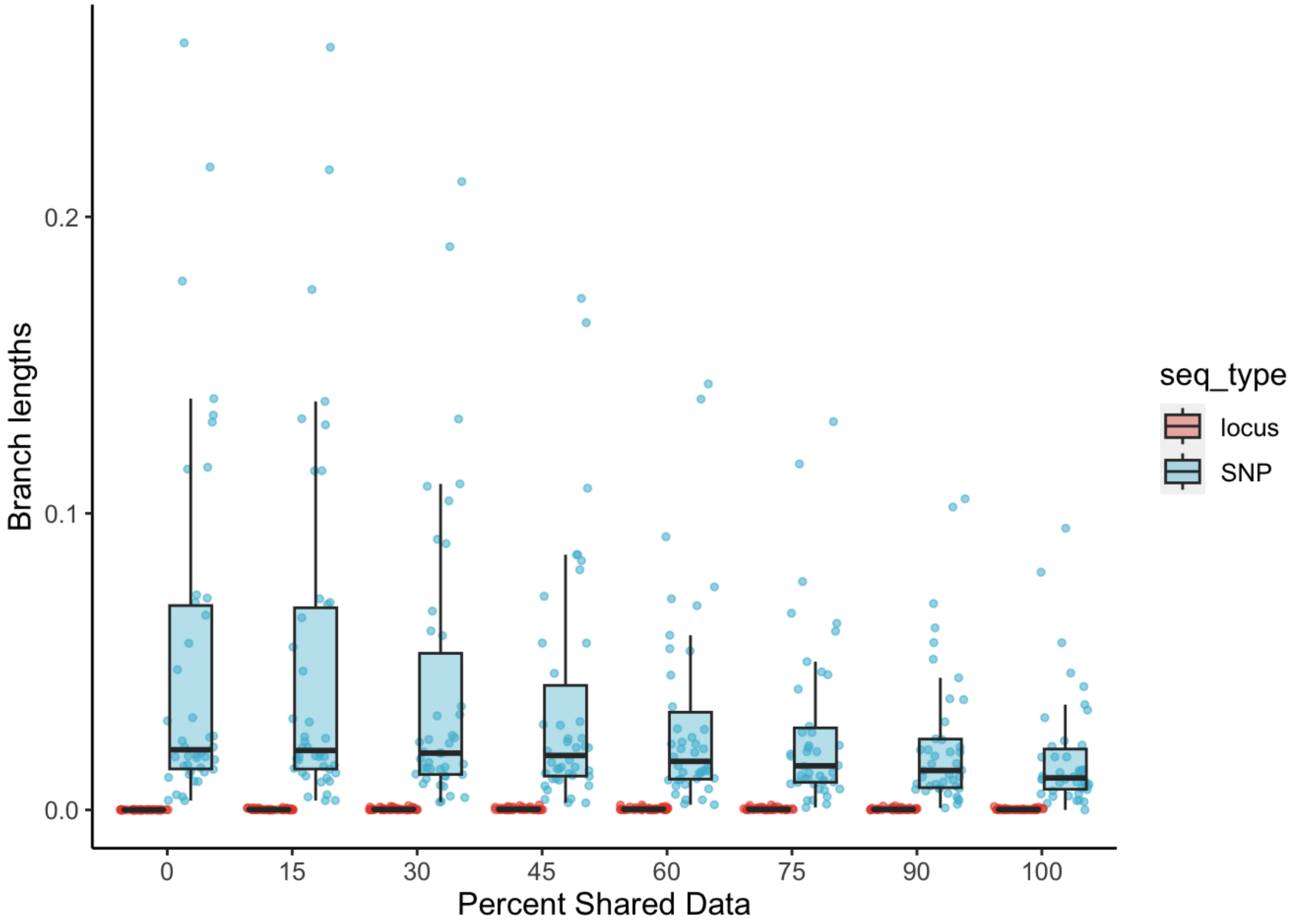
```r
alltreeFiles <-
  list.files(
    path = path,
    pattern = "*all.raxml.support",
    full.names = TRUE,
    recursive = FALSE
  )

alltree_list <- list()

for (i in 1:length(alltreeFiles)) {
  a <-
    paste(gsub("\\..*", "", basename(alltreeFiles[i])), ".tree", sep = "")
  tree <- read.tree(alltreeFiles[i])
  alltree_list[[a]] <- tree
}

#Same thing as above but for SNPS

snptreeFiles <-
  list.files(
    path = path,
    pattern = "*variant.raxml.support",
    full.names = TRUE,
    recursive = FALSE
  )

snptree_list <- list()

for (i in 1:length(snptreeFiles)) {
  a <-
    paste(gsub("\\..*", "", basename(snptreeFiles[i])), ".tree", sep = "")
  tree <- read.tree(snptreeFiles[i])
  snptree_list[[a]] <- tree

}
```

Analyze the data

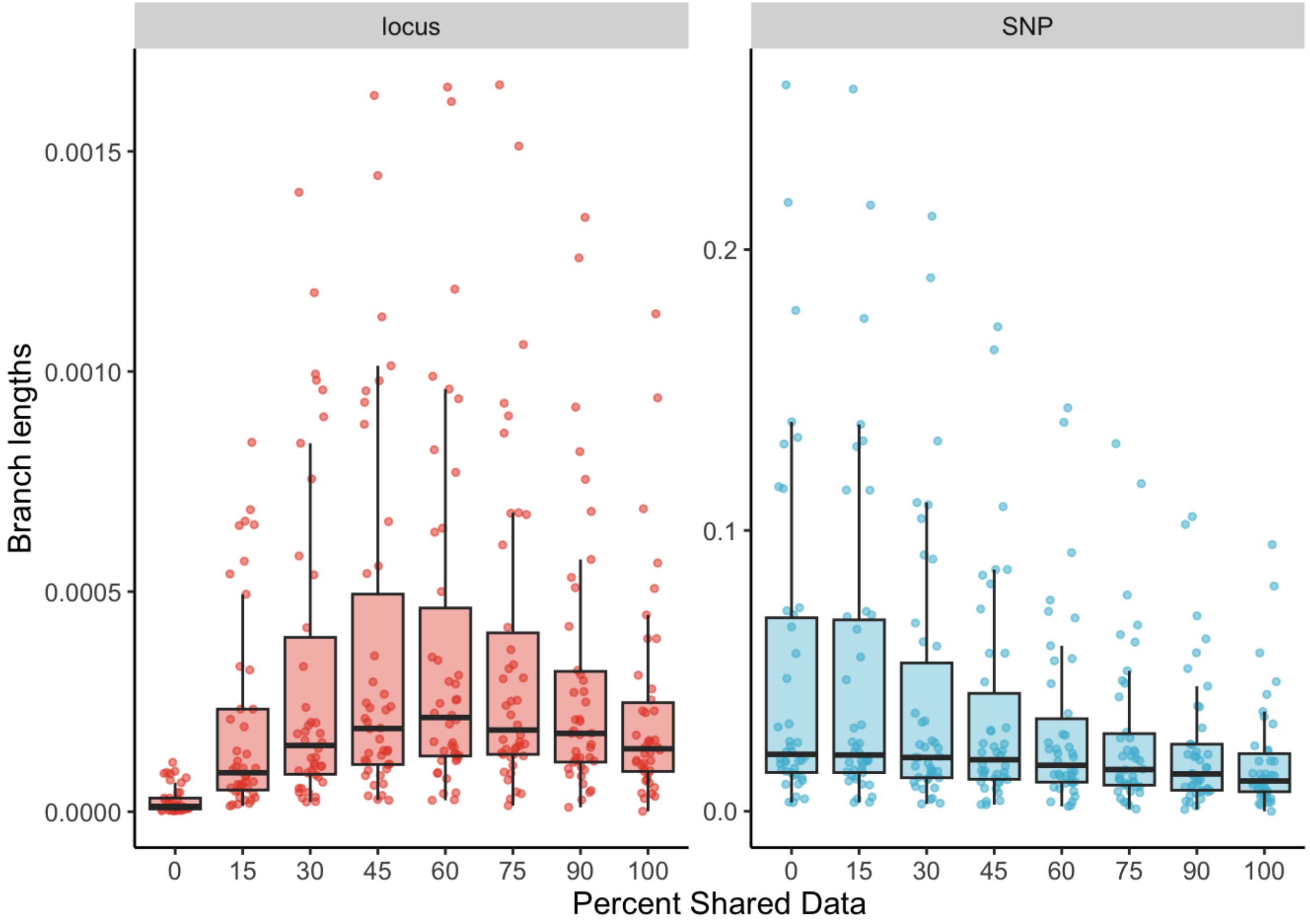
Extract all of the edge lengths and node label data from the trees

First make simple boxplots. Start with a boxplots of edgelength

```
Order<-c( "0", "15", "30", "45", "60", "75", "90", "100")

edgePlot<- ggplot(totTree1.dat, aes(y=edgeLength, x=factor(filter, levels = Order))) +geom_point(aes(color=seq_type, alpha=0.5), size= 1, show.legend = FALSE,  position=position_jitterdodge())+ geom_boxplot(aes(fill=seq_type), alpha=0.4, outlier.colour = NA)+ xlab("Percent Shared Data")+ ylab("Branch lengths")+ scale_fill_manual(values= pal_npg("nrc", alpha = 0.7)(2))+ scale_color_manual(values= pal_npg("nrc", alpha = 0.6)(2)) + theme(text = element_text(size=12)) + theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank(), axis.line = element_line(colour = "black"))

edgePlot
```

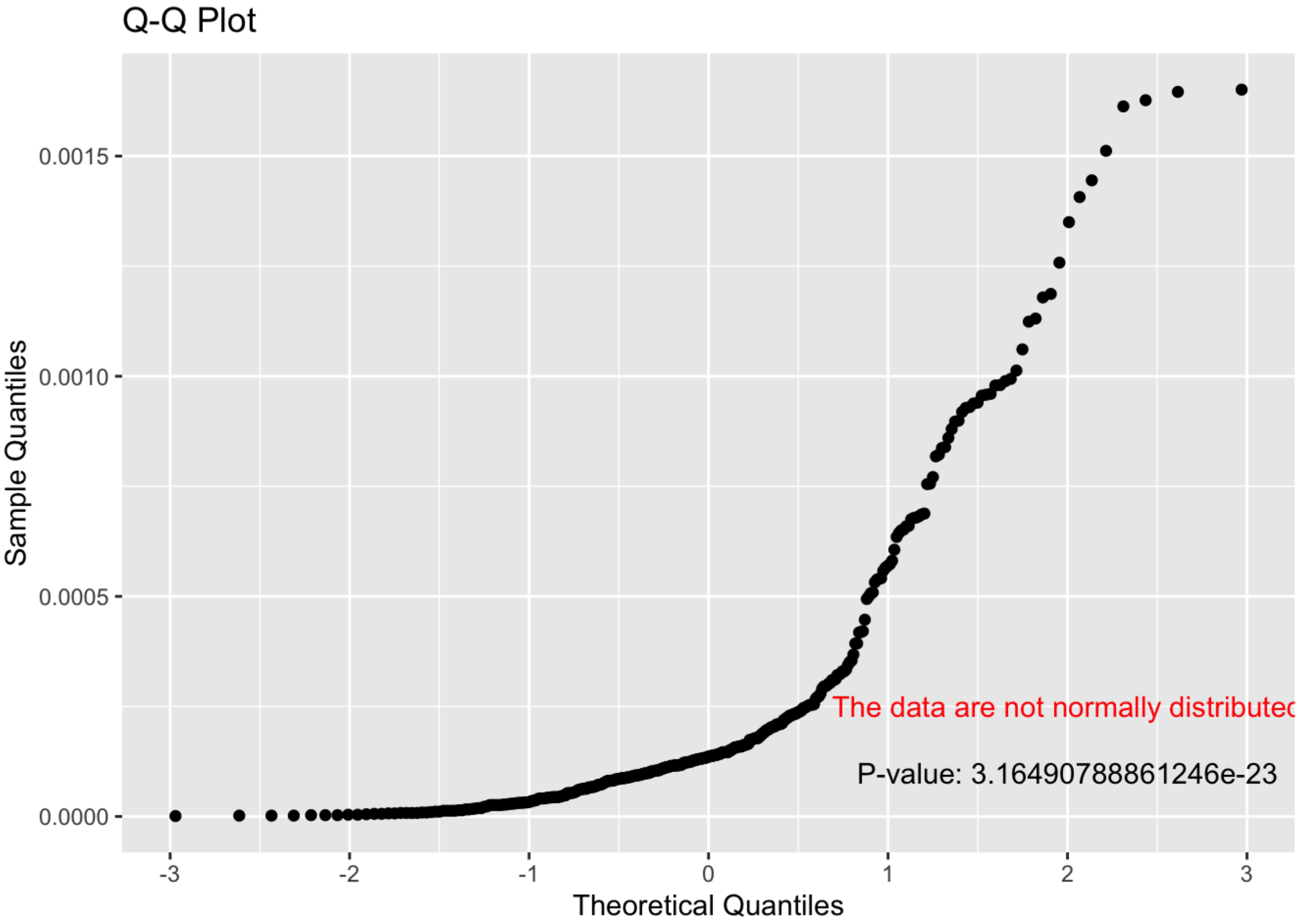


Plot branch lengths differently from edge lengths



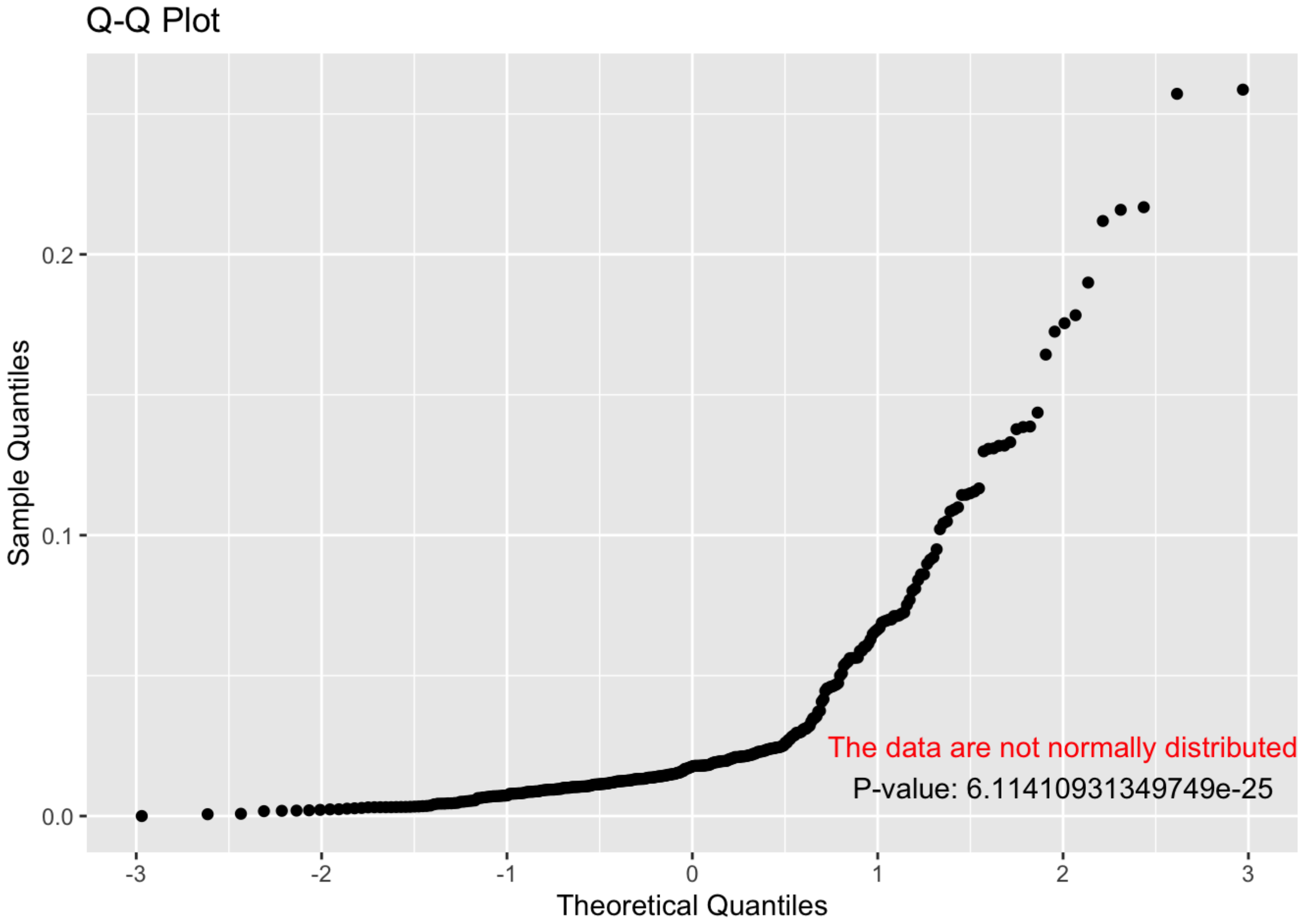
Test the normality of both locus branch length datasets

```
#Locus data
loc.dat <- totTree1.dat%>%
  filter(seq_type == "locus", .preserve = TRUE)

# Shapiro-Wilk Test for Normality
shapiro.test_result <- shapiro.test(loc.dat$edgeLength)
shapiro_p_value <- shapiro.test_result$p.value

# Q-Q Plot
qq_plot <- qqnorm(loc.dat$edgeLength, plot.it = FALSE)
qq_plot_data <- data.frame(Theoretical = qq_plot$x, Sample = qq_plot$y)
qq_plot <- ggplot(qq_plot_data, aes(Theoretical, Sample)) +
  geom_point()  +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
  ggtitle("Q-Q Plot") +
  annotate("text", x = 2, y = 0.0001, label = paste("P-value:", shapiro_p_value)) +
  annotate("text", x = 2, y = 0.00025, label = ifelse(shapiro_p_value < 0.01, "The data are not normally distributed", "The data are normally distributed"), color = ifelse(shapiro_p_value < 0.01, "red", "black"))

# Display the Q-Q plot with annotations
print(qq_plot)
```

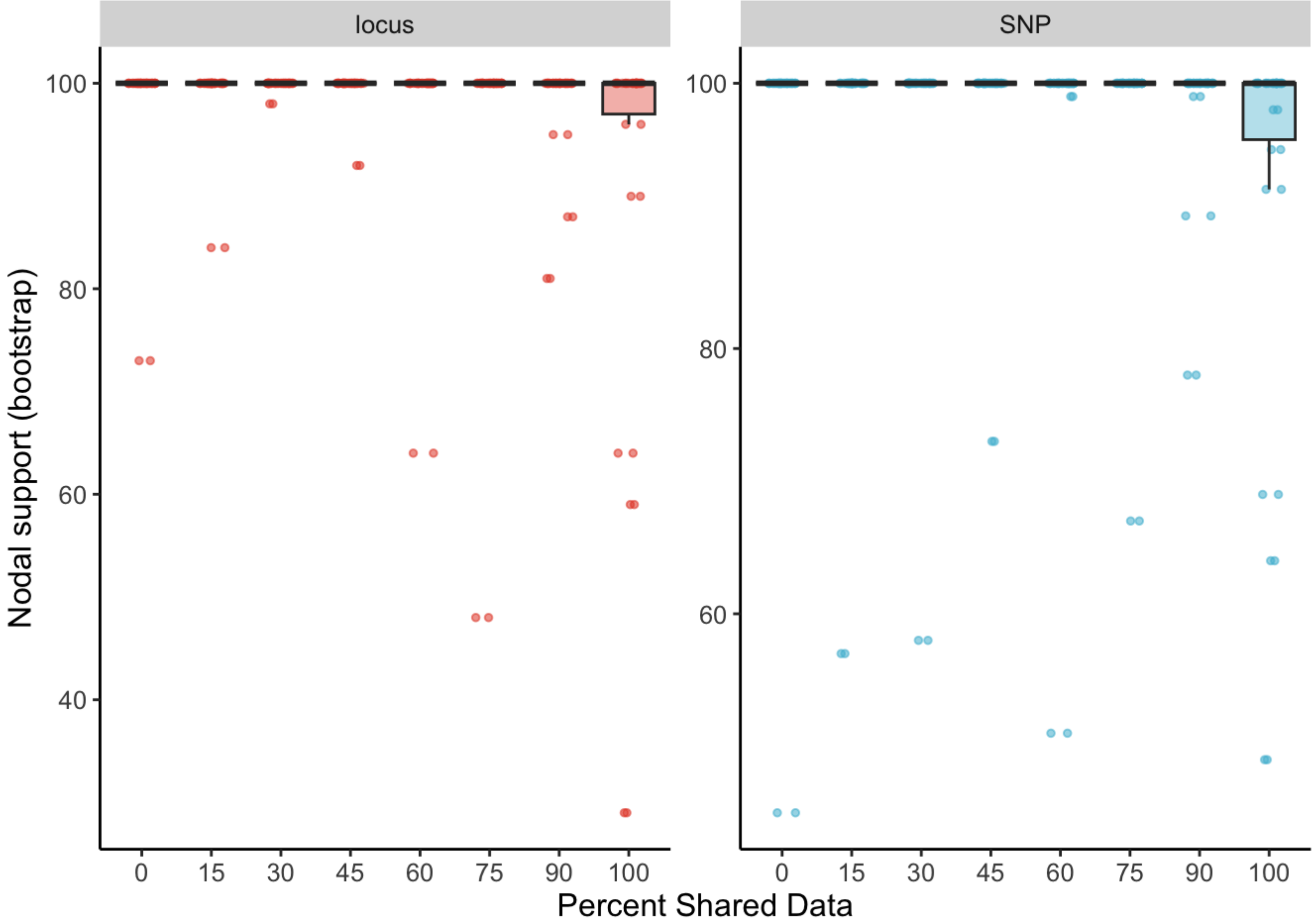


Test the normality of both snp branch length datasets

```
#SNP data
SNP.dat <- totTree1.dat%>%
  filter(seq_type == "SNP", .preserve = TRUE)

# Shapiro-Wilk Test for Normality
shapiro.test_result <- shapiro.test(SNP.dat$edgeLength)
shapiro_p_value <- shapiro.test_result$p.value

# Q-Q Plot
qq_plot <- qqnorm(SNP.dat$edgeLength, plot.it = FALSE)
qq_plot_data <- data.frame(Theoretical = qq_plot$x, Sample = qq_plot$y)
qq_plot <- ggplot(qq_plot_data, aes(Theoretical, Sample)) +
  geom_point() +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles") +
  ggtitle("Q-Q Plot") +
  annotate("text", x = 2, y = 0.01, label = paste("P-value:", shapiro_p_value)) +
  annotate("text", x = 2, y = 0.025, label = ifelse(shapiro_p_value < 0.01, "The data are not normally distributed", "The data are normally distributed"), color = ifelse(shapiro_p_value < 0.01, "red", "black"))

# Display the Q-Q plot with annotations
print(qq_plot)
```



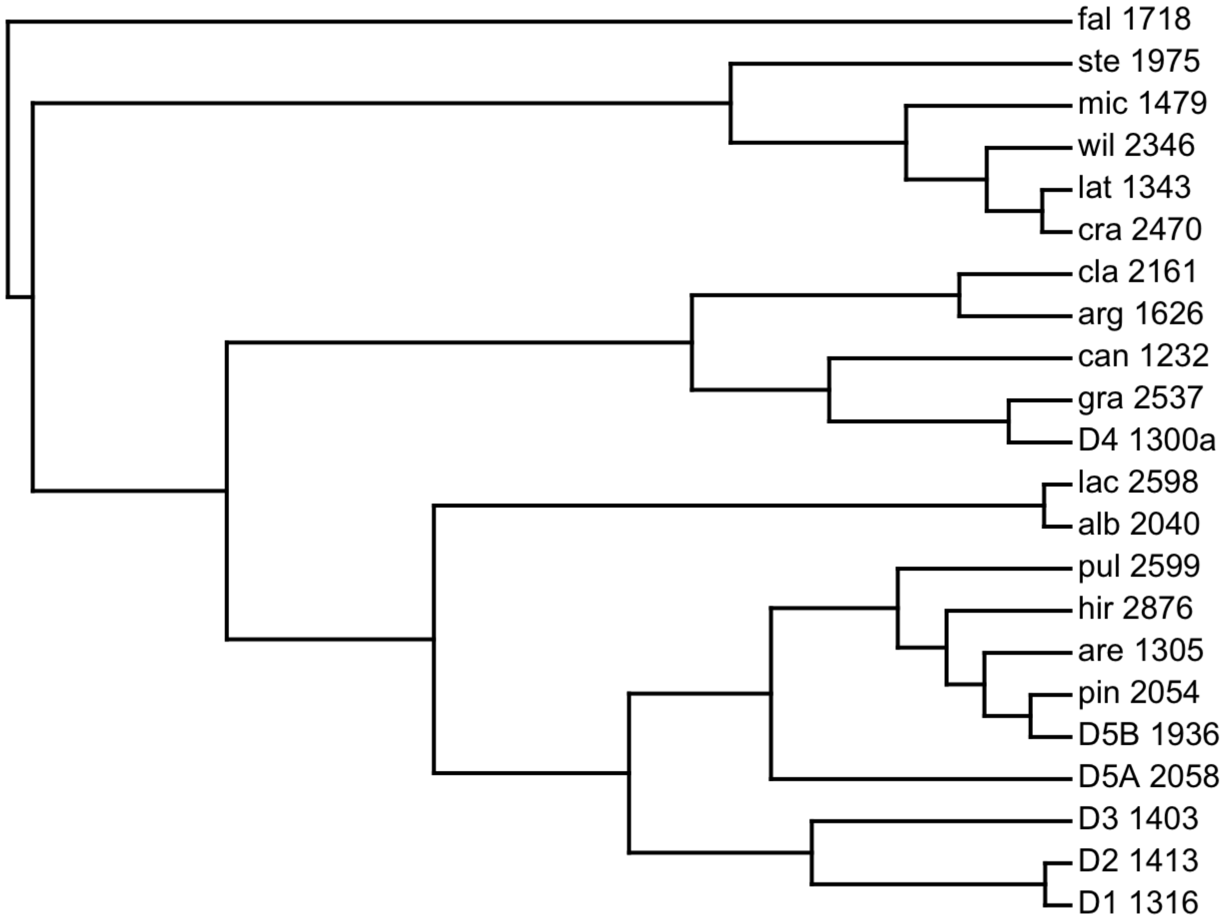
Make boxplots of nodal support



Explore relative branch lengths across each raxml tree. Bring in the BEAST data and explore patterns of node age across the different data and filtering types

Plot one of the beast trees to visualize

```
plotTree(allBeastTree_list[["30P_all_MCC.tre"]]@phylo)
```

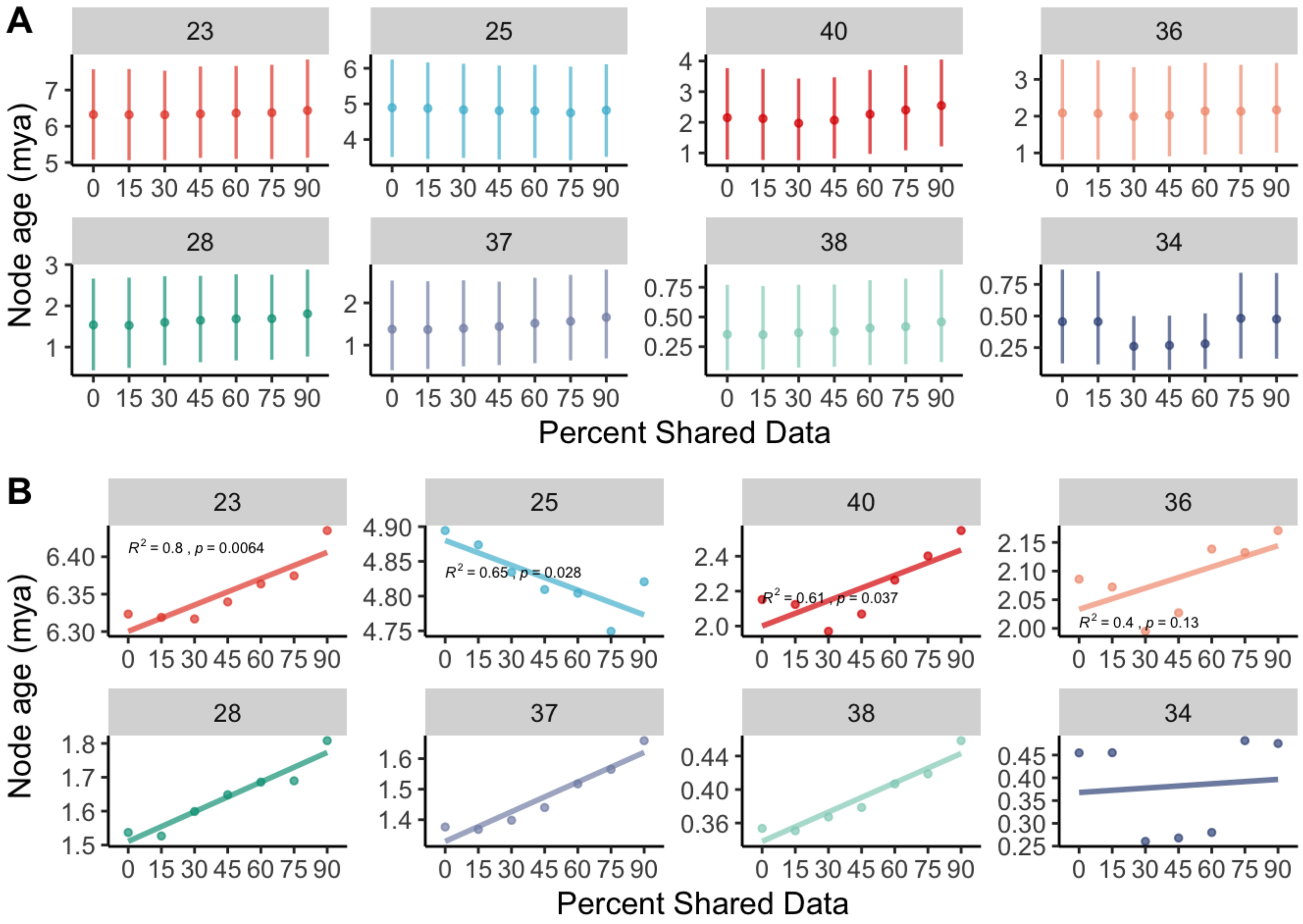


Extract all of the ages for the tree

Make a new columns where we specifically denote the percent missing data, and whether its variant or full locus

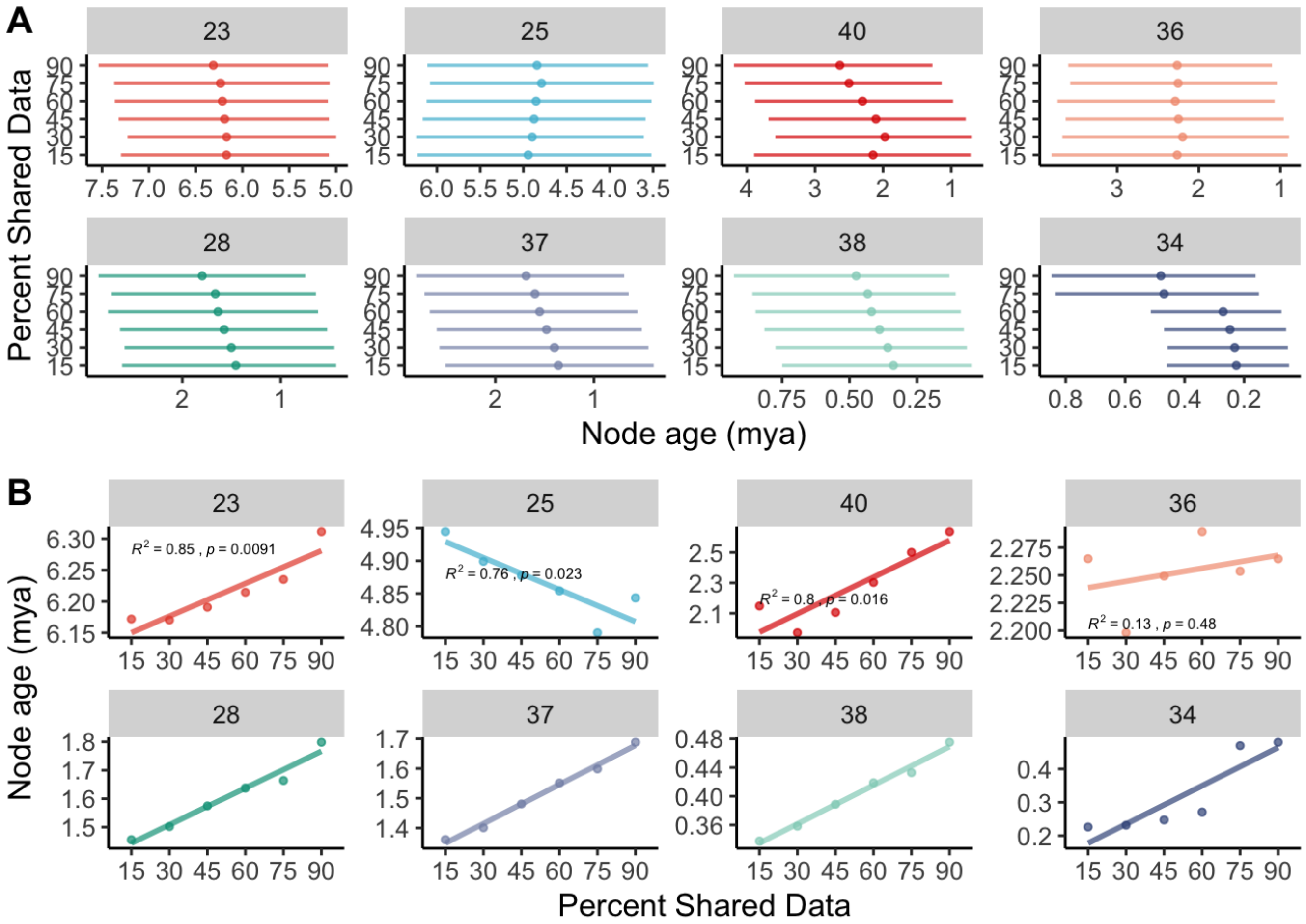
Plot the distributon of node ages as a function of percent missing data SNPS

```
snp_only <- totBeastTree1.dat%>%
  filter(seq_type == "SNP", .preserve = TRUE )%>%
  filter(!is.na(early), .preserve = TRUE)%>%
  mutate(node = as.numeric(node))%>%
  filter(node ==  "23"| node == "25"| node == "40" |node == "36" | node == "28" |node ==
"37" |node == "38"| node == "34" )%>%#Select a 8 nodes
  mutate(across(node, factor, levels=c("23"  , "25"  , "40"  , "36"   , "28"  , "37"  ,
"38"  , "34")))


  m <-ggplot(snp_only, aes(x= filtered, y=CAheight_mean, group = node )) +geom_point(aes
(color= as.character(node) ), position = position_jitter(height = 0, width = 0, seed =
1)  , size=1) + facet_wrap(~node, nrow = 2, scales = "free" )+ guides(color="none") +geo
m_segment( aes(y=early, yend=late, x=filtered, xend= filtered, color= as.character(nod
e)), position = position_jitter(height = 0, width = 0, seed = 1), size= 0.6) + guides(co
lor=FALSE) + ylab("Node age (mya)")+ xlab("Percent Shared Data")+ scale_color_manual(val
ues= pal_npg("nrc", alpha = 0.7)(8))+  theme(text = element_text(size=12)) + theme(pane
l.grid.major = element_blank(), panel.grid.minor = element_blank(),panel.background = el
ement_blank(), axis.line = element_line(colour = "black"))+facet_wrap(~node, nrow = 2, s
cales = "free" )


#Then plot just the mean and run a linear regression!

  n <-ggplot(snp_only, aes(x= filtered, y=CAheight_mean, group = node )) +geom_point(aes
(color= as.character(node) ), position = position_jitter(height = 0, width = 0, seed =
1)  , size=1) + facet_wrap(~node, nrow = 2, scales = "free" )+ guides(color="none")+ geo
m_smooth(method = "lm", aes(color = as.character(node) ), se =FALSE )+   stat_cor(aes(la
bel = paste(..rr.label.., ..p.label.., sep = "~`,`~")), size=2)  + guides(color=FALSE) +
ylab("Node age (mya)")+ xlab("Percent Shared Data")+ scale_color_manual(values= pal_npg
("nrc", alpha = 0.7)(8))+  theme(text = element_text(size=12)) + theme(panel.grid.major
= element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank
(), axis.line = element_line(colour = "black"))+facet_wrap(~node, nrow = 2, scales = "fr
ee" )



ggarrange(m, n ,
          labels = c("A", "B"),
           nrow =2, widths = 8, heights = 5)
```



Plot the distributon of node ages as a function of percent missing data Locus

```
all_only <- totBeastTree1.dat%>%
  filter(seq_type == "locus", .preserve = TRUE )%>%
 filter(!is.na(early), .preserve = TRUE)%>%
  mutate(node = as.numeric(node))%>%
  filter(node ==  "23"| node == "25"| node == "40" |node == "36" | node == "28" |node ==
"37" |node == "38"| node == "34" )%>%#Select a 8 nodes
  mutate(across(node, factor, levels=c("23"  , "25"  , "40"  , "36"   , "28"  , "37"  ,
"38"  , "34")))


  l<- ggplot(all_only, aes(y= filtered, x=CAheight_mean )) +geom_point(aes(color= as.cha
racter(node) ),   position = position_jitter(height = 0, width = 0, seed = 1)  , size=1)
+geom_segment( aes(x=early, xend=late, y=filtered, yend= filtered, color= as.character(n
ode)), position = position_jitter(height = 0, width = 0, seed = 1), size= 0.6)+scale_x_r
everse() + guides(color=FALSE) + xlab("Node age (mya)")+ ylab("Percent Shared Data")+ sc
ale_color_manual(values= pal_npg("nrc", alpha = 0.7)(8))+ facet_wrap(~node, nrow = 2, sc
ales = "free" )+ theme(text = element_text(size=12)) + theme(panel.grid.major = element_
blank(), panel.grid.minor = element_blank(),panel.background = element_blank(), axis.lin
e = element_line(colour = "black"))


  p<-   ggplot(all_only, aes(x= filtered, y=CAheight_mean, group = node )) +geom_point(a
es(color= as.character(node) ), position = position_jitter(height = 0, width = 0, seed =
1)  , size=1) + facet_wrap(~node, nrow = 2, scales = "free" )+ guides(color="none")+ geo
m_smooth(method = "lm", aes(color = as.character(node) ), se =FALSE )+   stat_cor(aes(la
bel = paste(..rr.label.., ..p.label.., sep = "~`,`~")), size=2)  + guides(color=FALSE) +
ylab("Node age (mya)")+ xlab("Percent Shared Data")+ scale_color_manual(values= pal_npg
("nrc", alpha = 0.7)(8))+  theme(text = element_text(size=12)) + theme(panel.grid.major
= element_blank(), panel.grid.minor = element_blank(),panel.background = element_blank
(), axis.line = element_line(colour = "black"))+facet_wrap(~node, nrow = 2, scales = "fr
ee" )



ggarrange(l, p ,
          labels = c("A", "B"),
           nrow =2, widths = 8, heights = 5)
```



Phylogenetic comparative methods

Use trees from the 45% filtered trees to explore the impact of SNP vs. locus datasets and the use of phylograms vs. chronograms in PCMs.

First create a funciton that trims data and puts it into the right format for stochastic character mapping

Run stochastic character mapping on the randomized data

```
#import data
snp_mock.dat<-read.csv(path2)

#bring in the three trees of interest

locus_45raxml.tre<-read.tree(path3)

snp_45raxml.tre<-read.tree(path4)

locus_45beast.tre<-read.beast(path5)

locus_45beast.tre <- as.phylo(locus_45beast.tre)

snp_45beast.tre<-read.beast(path6)

snp_45beast.tre<-as.phylo(snp_45beast.tre)


trees.list <- c(locus_45raxml.tre,snp_45raxml.tre,locus_45beast.tre, snp_45beast.tre  )

trees.list.name <- c("locus_45raxml","snp_45raxml","locus_45beast", "snp_45beast"  )
```

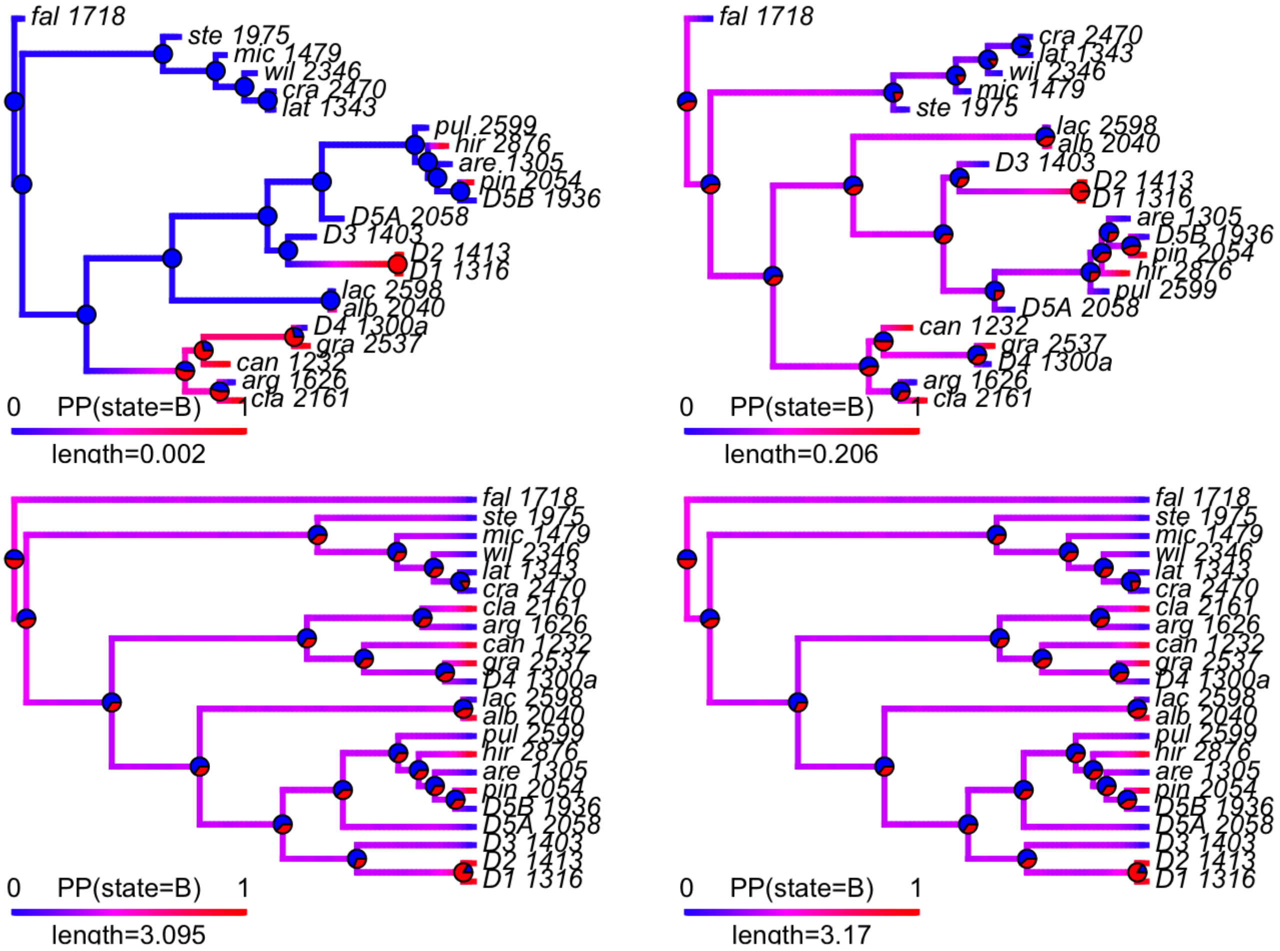
Loop through each tree and run a stochastic character map and density map

```
mockdat.mode<-setNames(snp_mock.dat$State,snp_mock.dat$Tip)

mockdat.mode<-as.factor(mockdat.mode) #This is essential now for some rason...

simmap_list <- list()
dmap_list <-list()
summary_list<-list()
for (i in 1:length(trees.list)) {

  a <-make.simmap(tree = trees.list[i], x =mockdat.mode, model = "ARD", nsim = 1000, Q=
"empirical" )

  tree_file <- paste0( trees.list.name[i])
  simmap_list[[tree_file]] <- a

  simmap_list[[i]] <- a


  b <- densityMap(a,plot=FALSE)

  dmap_list[[tree_file]] <- b

  c<- summary(a)

  summary_list[[tree_file]]<-c

}
```

```
## make.simmap is sampling character histories conditioned on
## the transition matrix
##
## Q =
##         A       B
## A -0.4719  0.4719
## B  0.4719 -0.4719
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##   A   B
## 0.5 0.5
## make.simmap is sampling character histories conditioned on
## the transition matrix
##
## Q =
##           A         B
## A -25.55625  25.55625
## B  41.28740 -41.28740
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##   A   B
## 0.5 0.5
## make.simmap is sampling character histories conditioned on
## the transition matrix
##
## Q =
##           A         B
## A -2.240632  2.240632
## B  4.063175 -4.063175
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##   A   B
## 0.5 0.5
## make.simmap is sampling character histories conditioned on
## the transition matrix
##
## Q =
##           A         B
## A -2.352308  2.352308
## B  4.300970 -4.300970
## (estimated using likelihood);
## and (mean) root node prior probabilities
## pi =
##   A   B
## 0.5 0.5
```

Plot the simmaps

```
par(mfrow = c(2, 2))  # Set up a 2x2 grid layout for the plots



for (i in 1:length(dmap_list)) {

  # Extract the file name
  file_name <- names(dmap_list)[i]


 plot(dmap_list[[file_name]])+ nodelabels(node = as.numeric(row.names(summary_list[[file_name]]$ace)), pie=summary_list[[file_name]]$ace, piecol = c("blue", "red"), cex=0.7 )


}
```



Run ancestral character estimation on a continuous character

Simulate a continuous character based on brownian motion on the 45 filtered SNP Beast tree

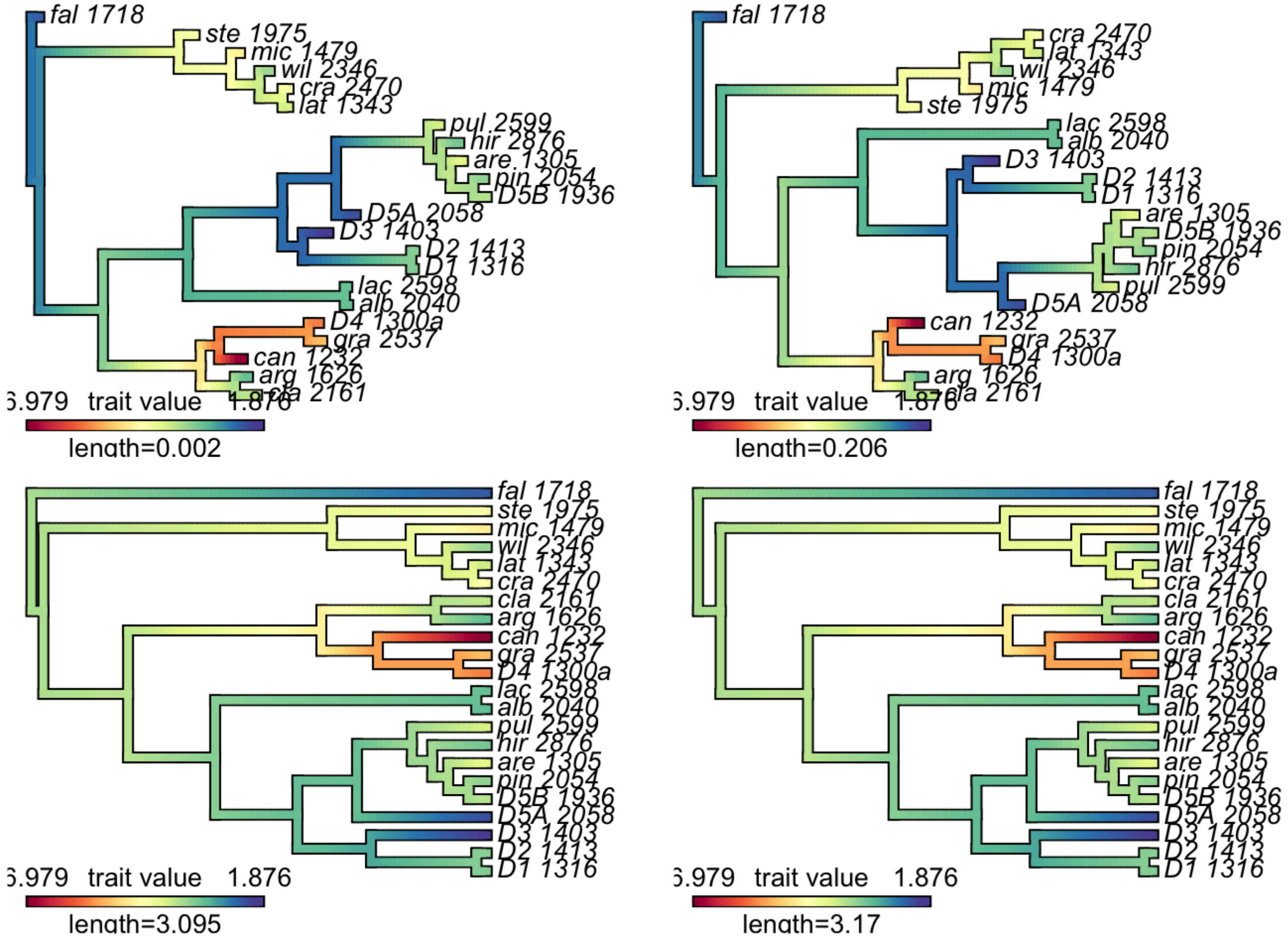
```
# Set the parameters for the Brownian motion simulation
sigma <- 1  # Trait evolutionary rate
nsim <- 1  # Number of simulations

# Simulate the continuous trait using Brownian motion
sim_data <- fastBM(snp_45beast.tre, sigma = sigma, nsim = nsim)
```

Loop through and make cont maps

```
#Make anc recons TEST
fitBM<-anc.ML(snp_45beast.tre,sim_data,model="BM")



BM<-contMap(snp_45beast.tre,sim_data, method="user",anc.states=fitBM$ace,plot=FALSE)

#Make them for all the trees

fit_list <-list()
contmap_list <- list()

for (i in 1:length(trees.list)) {

  a <-anc.ML(tree = trees.list[[i]], x =sim_data, model = "BM" )

  tree_file <- paste0( trees.list.name[i])
  fit_list[[tree_file]] <- a

  fit_list[[i]] <- a

  b<-contMap(trees.list[[i]],sim_data, method="user",anc.states=a$ace,plot=FALSE)

  b <-setMap(b, brewer.pal(11,"Spectral"))


  contmap_list[[tree_file]] <- b


}
```

Plot the contmaps and save everything

```
par(mfrow = c(2, 2))  # Set up a 2x2 grid layout for the plots



for (i in 1:length(dmap_list)) {


  # Extract the file name
  file_name <- names(contmap_list)[i]

 plot(contmap_list[[file_name]])

}
```



#Done! If you. have any questions please email Jacob Suissa at jsuissa@utk.edu (mailto:jsuissa@utk.edu)