Microbial Innovations

Mario E. Muscarella, James P. O'Dwyer

Last updated on 16 September, 2016

Initial Setup

```
# Clear Environment and Set Working Directory
rm(list=ls())
setwd("~/GitHub/microbial-innovations/analyses")

# Add Basic Functions
se <- function(x, ...){sd(x, na.rm = TRUE)/sqrt(length(na.omit(x)))}
ci <- function(x, ...){1.96 * sd(x,na.rm = TRUE)}

# Load Packages
require("ape"); require("png"); require("grid")
```

```
## Loading required package: ape

## Loading required package: png

## Loading required package: grid
```

```
require("ggtree")
```

```
## Loading required package: ggtree

## Loading required package: ggplot2

## If you use ggtree in published research, please cite:
##
## Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam.
## ggtree: an R package for visualization and annotation of phylogenetic trees with their covariates an
## Methods in Ecology and Evolution 2016, doi:10.1111/2041-210X.12628

##
## Attaching package: 'ggtree'

## The following object is masked from 'package:ape':
##
##     rotate
```

```
require("phytools")
```

```
## Loading required package: phytools

## Loading required package: maps

##
## Attaching package: 'phytools'

## The following object is masked from 'package:ggtree':
##
##     reroot
```

```
library("colorspace")
```

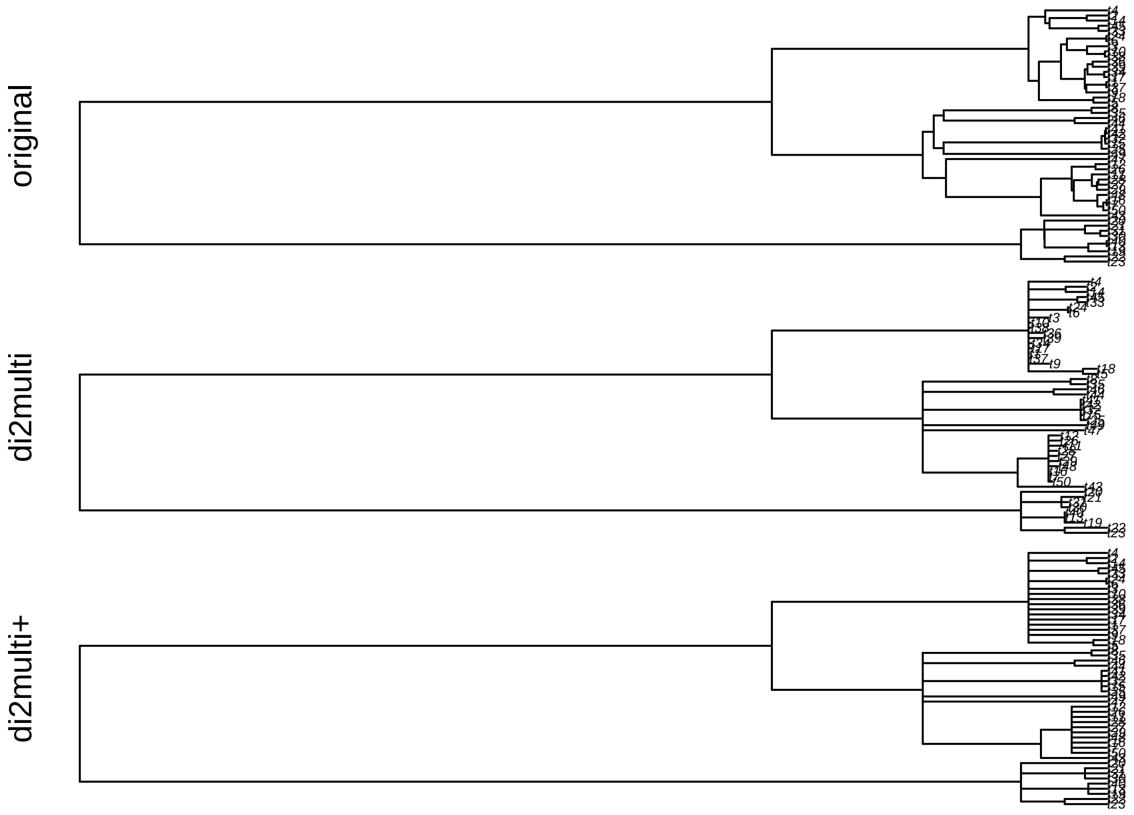
```
# Load Source Functions
source("../bin/MothurTools.R")
```

```
## Loading required package: reshape
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:ggtree':
##
##     expand
```

```
source("../bin/di2multi2.R")
```



```
# Define Simple Functions
ggplotColours <- function(n = 6, h = c(0, 360) + 15){
  if ((diff(h) %% 360) < 1) h[2] <- h[2] - 360/n
  hcl(h = (seq(h[1], h[2], length = n)), c = 100, l = 65)
}

ttest <- function(reg, coefnum, val){
  co <- coef(summary(reg))
  tstat <- (co[coefnum,1]-val)/co[coefnum,2]
  pstat <- 2 * pt(abs(tstat), reg$df.residual, lower.tail = FALSE)
  return(list = c(t = tstat, df = reg$df.residual, p =  pstat))
}

# Save Default Plot Settings
opar <- par(no.readonly = TRUE)  # Saves plot defaults
```

Import Tree Files

```
LTP <- read.tree("../data/LTPs123.tree")
NTL <- read.tree("../data/NTL.tree")
RDP <- read.tree("../data/RDP.tree")
```

Import Taxonomy Information

```
LTP.tax <- read.tax("../data/LTPs123_unique.pds.wang.taxonomy", format = "rdp", col.tax = 2)
NTL.tax <- read.tax("../data/nmicrobiol201648_s7.pds.wang.taxonomy", format = "rdp", col.tax = 2)
RDP.tax <- read.tax("../data/rdp_download_9752seqs.pds.wang.taxonomy", format = "rdp", col.tax = 2)
```

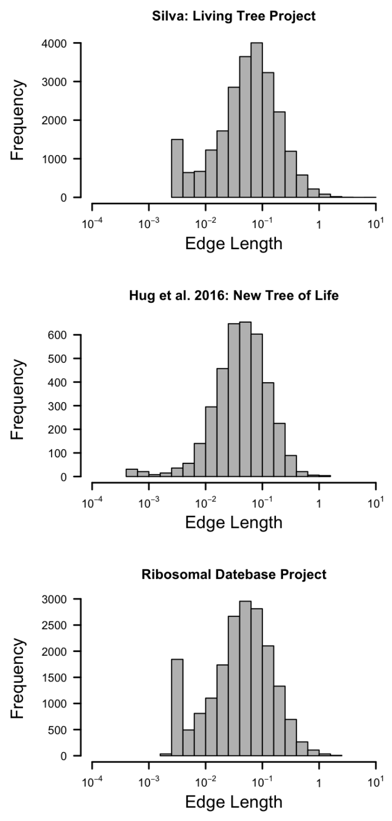
Edge Length Distribution

```
LTP.edges <- LTP$edge.length
NTL.edges <- NTL$edge.length
RDP.edges <- RDP$edge.length

png(filename="../figures/BranchLengths.png",
    width = 800, height = 1600, res = 96*2)

layout(matrix(1:3, 3, 1))
par(mar = c(5, 6, 3, 1) + 0.1, oma = c(1,1,1,1))

hist(log10(LTP.edges), main = "Silva: Living Tree Project",
     axes = F, xlab = "", ylab = "", xlim = c(-4, 1), col = "gray")

axis(1, at = c(-4, -3, -2, -1, 0, 1),
     labels = expression(10^-4, 10^-3, 10^-2, 10^-1, 1, 10^1),
     las = 1, lwd = 1.5)
axis(2, labels = T, las = 1, lwd = 1.5)

mtext("Edge Length", side = 1, cex = 1, line = 2.5)
mtext("Frequency", side = 2, cex = 1, line = 4)

hist(log10(NTL.edges), main = "Hug et al. 2016: New Tree of Life",
     axes = F, xlab = "", ylab = "", xlim = c(-4, 1), col = "gray")

axis(1, at = c(-4, -3, -2, -1, 0, 1),
     labels = expression(10^-4, 10^-3, 10^-2, 10^-1, 1, 10^1),
     las = 1, lwd = 1.5)
axis(2, labels = T, las = 1, lwd = 1.5)

mtext("Edge Length", side = 1, cex = 1, line = 2.5)
mtext("Frequency", side = 2, cex = 1, line = 4)

hist(log10(RDP.edges), main = "Ribosomal Datebase Project",
     axes = F, xlab = "", ylab = "", xlim = c(-4, 1), col = "gray")
```

```
axis(1, at = c(-4, -3, -2, -1, 0, 1),
     labels = expression(10^-4, 10^-3, 10^-2, 10^-1, 1, 10^1),
     las = 1, lwd = 1.5)
axis(2, labels = T, las = 1, lwd = 1.5)

mtext("Edge Length", side = 1, cex = 1, line = 2.5)
mtext("Frequency", side = 2, cex = 1, line = 4)


# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/BranchLengths.png")
grid.raster(img)
```



Edge Length vs Support

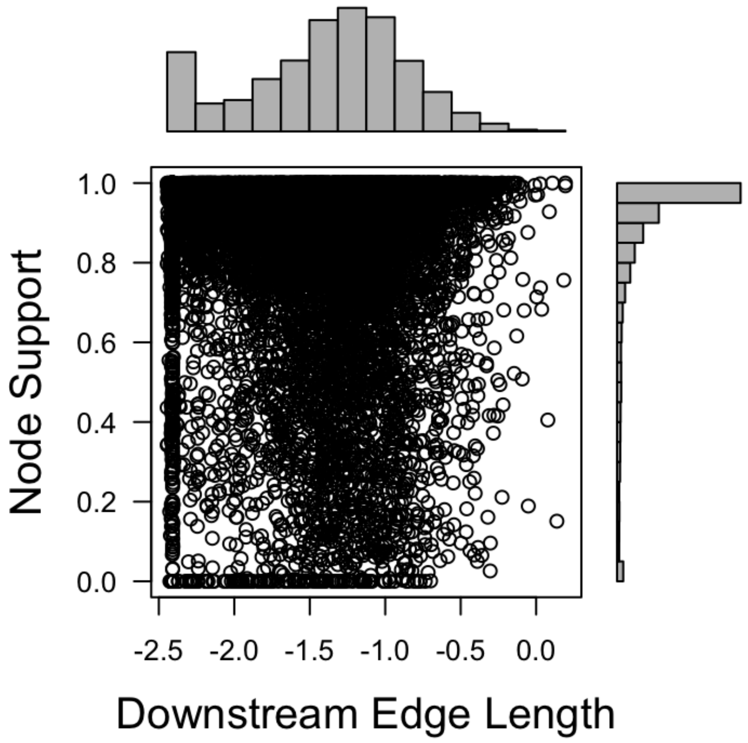
```
node.tab <- data.frame(matrix(NA, LTP$Nnode, 5))
colnames(node.tab) <- c("node", "support", "edge_up", "edge_down1", "edge_down2")
```

```r
node.tab$node <- as.numeric(c(1:LTP$Nnode))
node.tab$support <- LTP$node.label
node.tab[1,3] <- NA
for (i in 2:LTP$Nnode){
  edge.up <- which(LTP$edge[,2] == length(LTP$tip.label) + i)
  node.tab[i,3] <- LTP$edge.length[edge.up]
  edge.down <- which(LTP$edge[,1] == length(LTP$tip.label) + i)
  node.tab[i,4] <- LTP$edge.length[edge.down[1]]
  node.tab[i,5] <- LTP$edge.length[edge.down[2]]
}
node.tab$edge_down_min <- apply(cbind(node.tab$edge_down1, node.tab$edge_down2), 1, min)

scatterhist = function(x, y, xlab="", ylab=""){
  zones=matrix(c(2,0,1,3), ncol=2, byrow=TRUE)
  par(oma = c(1,1,1,1))
  layout(zones, widths=c(4/5,1/5), heights=c(1/5,4/5))
  xhist = hist(x, plot=FALSE)
  yhist = hist(y, plot=FALSE)
  top = max(c(xhist$counts, yhist$counts))
  par(mar=c(5,5,1,1))
  plot(x,y, las = 1, xlab = "", ylab = "")
  mtext(xlab, side=1, line=3, outer=F, cex = 1.25)
  mtext(ylab, side=2, line=3, outer=F, cex = 1.25)
  par(mar=c(0,5,1,1))
  barplot(xhist$counts, axes=FALSE, space=0)
  par(mar=c(5,0,1,1))
  barplot(yhist$counts, axes=FALSE, space=0, horiz=TRUE)
}
```

```r
png(filename="../figures/NodeSupport.png",
    width = 800, height = 800, res = 96*2)

# Upstream
scatterhist(x = log10(node.tab$edge_up), y = as.numeric(node.tab$support),
            xlab = "Upstream Edge Length", ylab = "Node Support")

# Downstream
scatterhist(log10(node.tab$edge_down_min), as.numeric(node.tab$support),
            xlab = "Downstream Edge Length", ylab = "Node Support")

mean(as.numeric(node.tab$support), na.rm = T)
mean(as.numeric(node.tab$support[log10(node.tab$edge_up) < -1.75]), na.rm = T)
mean(as.numeric(node.tab$support[log10(node.tab$edge_up) > 0.75]), na.rm = T)
mean(as.numeric(node.tab$support[log10(node.tab$edge_up) < -2.25]), na.rm = T)

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/NodeSupport.png")
grid.raster(img)
```

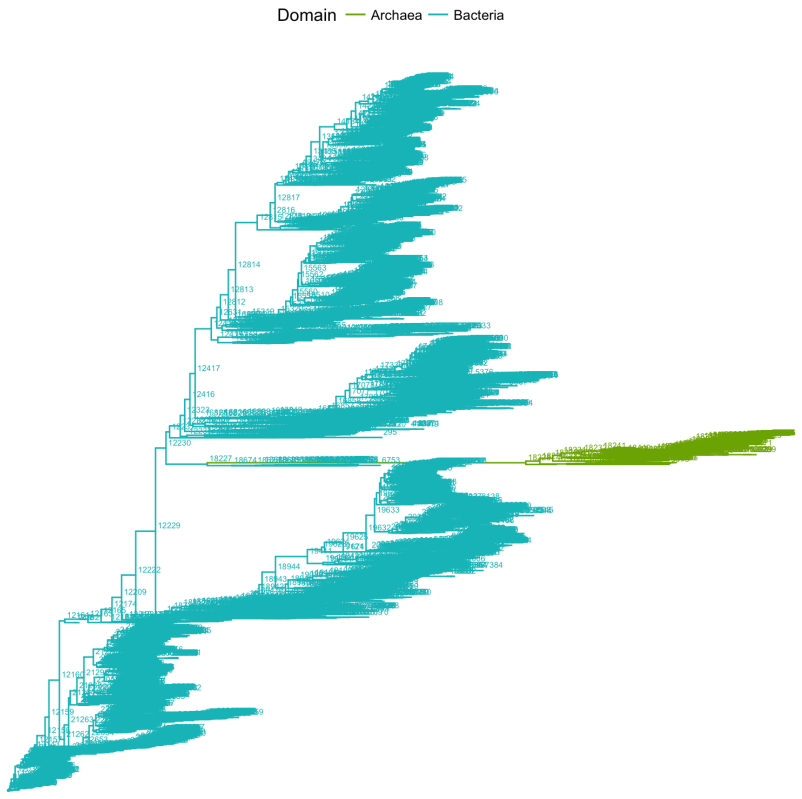


LTP Exploratory Tree with Domain Grouping

```
png(filename="../figures/LTP_explore.png",
    width = 1600, height = 1600, res = 96*2)

LTP.2 <- LTP
LTP.tax2 <- LTP.tax[match(LTP.2$tip.label, LTP.tax$OTU), ]
groupInfo <- split(LTP.2$tip.label, LTP.tax2$Domain)
livingTree <- groupOTU(LTP.2, groupInfo, group_name = "Domain")
levels(attributes(livingTree)$Domain) <- names(groupInfo)
ggtree(livingTree, aes(color = Domain), layout="rectangular") +
  geom_text(aes(subset=!isTip, label = node), show.legend = FALSE,
            hjust=-0.1, vjust = -0.5, size = 2) +
  theme(legend.position="top", legend.key = element_rect(colour = NA)) +
  scale_color_manual(values=c(ggplotColours(n = 4)[2:3]))

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/LTP_explore.png")
grid.raster(img)
```

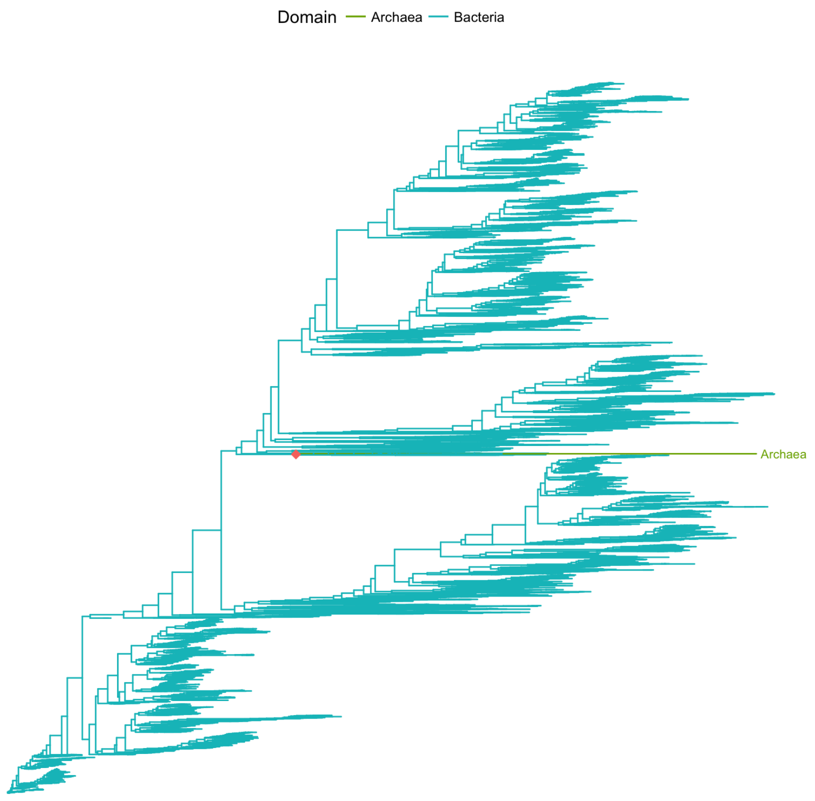


Identify the LTP Node that seperates bacteria and archaea

```
png(filename="../figures/LTP_archaea.png",
    width = 1600, height = 1600, res = 96*2)
par(mar=c(1,1,1,3))
cp <- ggtree(livingTree, aes(color = Domain), layout="rectangular") %>% collapse(node=18228)
cp + geom_point2(aes(subset=(node == 18227)), size=3, shape=18,
                 colour=ggplotColours(n = 4)[1], fill = "gray80") +
  theme(legend.position="top", legend.key = element_rect(colour = NA)) +
  geom_text2(aes(subset=(node == 18228), label = "Archaea"),
             show.legend = FALSE, hjust=-0.1, size = 3) +
  guides(colour = guide_legend(override.aes = list(shape = NA, label = NA))) +
  scale_color_manual(values=c(ggplotColours(n = 4)[2:3]))

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/LTP_archaea.png")
grid.raster(img)
```

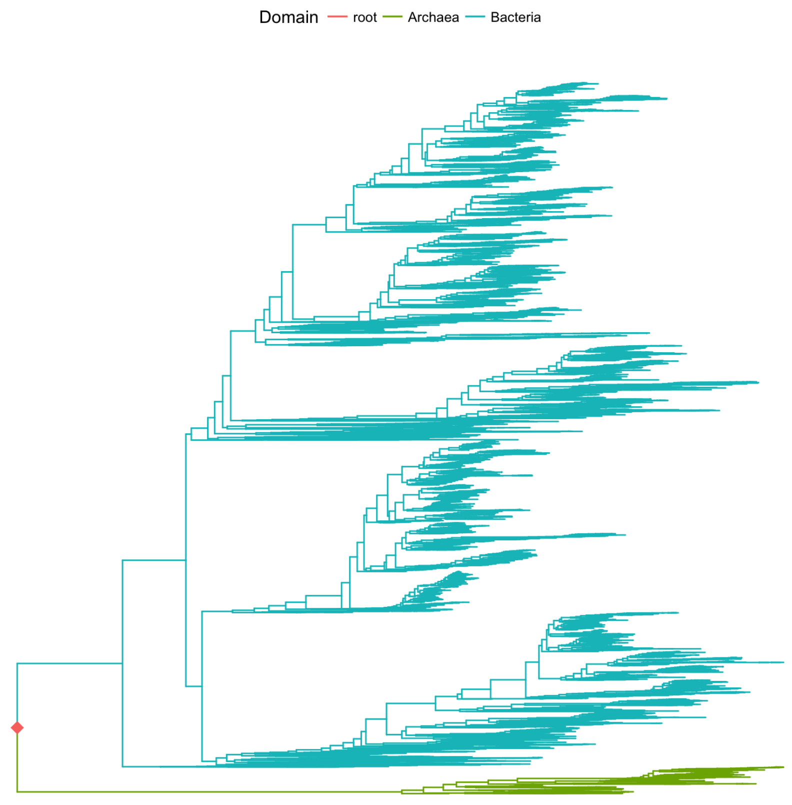


Reroot LTP Tree

```
png(filename="../figures/LTP_root.png",
    width = 1600, height = 1600, res = 96*2)

livingTree.r <- midpoint.root(livingTree)
livingTree.r2 <- groupOTU(livingTree.r, groupInfo, group_name = "Domain")
levels(attributes(livingTree.r2)$Domain) <- c("root", names(groupInfo))
rt <- ggtree(livingTree.r2, aes(color = Domain), layout="rectangular")
rt + geom_point2(aes(subset=(node == 11934)), size=4, shape=18,
                 colour=ggplotColours(n = 4)[1], fill = "gray80") +
  theme(legend.position="top", legend.key = element_rect(colour = NA)) +
  scale_color_manual(values=c(ggplotColours(n = 4)[1:3]))

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/LTP_root.png")
grid.raster(img)
```



Export Rooted Tree

```
write.tree(livingTree.r, file = "../data/LTP.rooted.tree")
```

Make Tree Ultrametric

I used treePL to make the tree ultrametric. My config file was pretty simple. It took way too long to get treePL to work. Maybe an alternative would be better. It looks like there are some program in R. The most common are chronos, chronopl, chronoMPL. But I don't know how they compare.

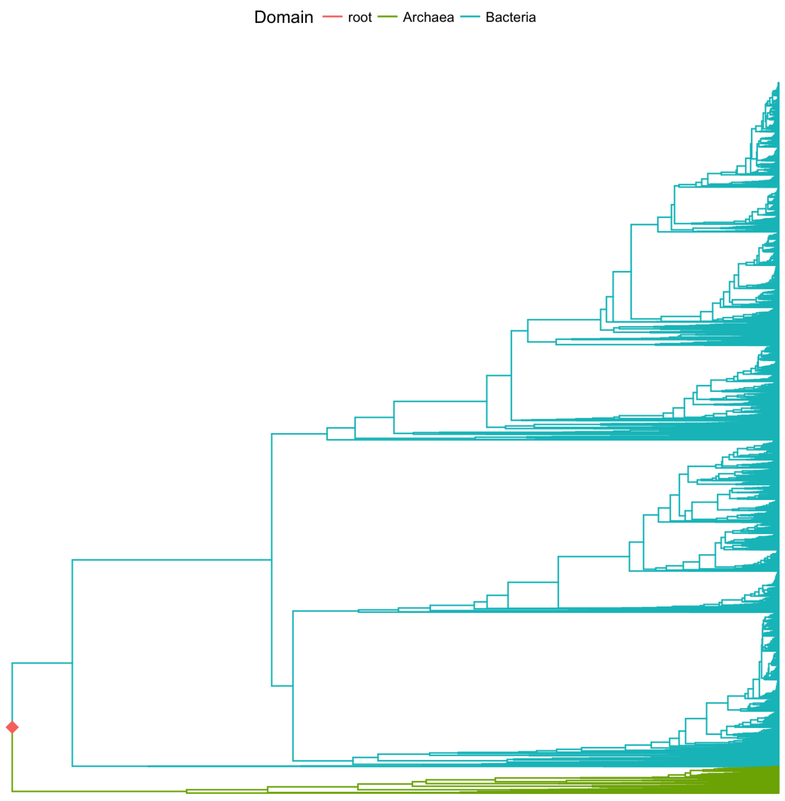
The output file was `LTP.dated.tree`

```
LTP.um <- read.tree("../data/LTP.dated.tree")

png(filename="../figures/LTP_dated.png",
    width = 1600, height = 1600, res = 96*2)

livingTree.um <- groupOTU(LTP.um, groupInfo, group_name = "Domain")
levels(attributes(livingTree.um)$Domain) <- c("root", names(groupInfo))
```

```r
um <- ggtree(livingTree.um, aes(color = Domain), layout="rectangular")
um + geom_point2(aes(subset=(node == 11934)), size=4, shape=18,
                 colour=ggplotColours(n = 4)[1], fill = "gray80") +
  theme(legend.position="top", legend.key = element_rect(colour = NA)) +
  scale_color_manual(values=c(ggplotColours(n = 4)[1:3]))

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```r
img <- readPNG("../figures/LTP_dated.png")
grid.raster(img)
```



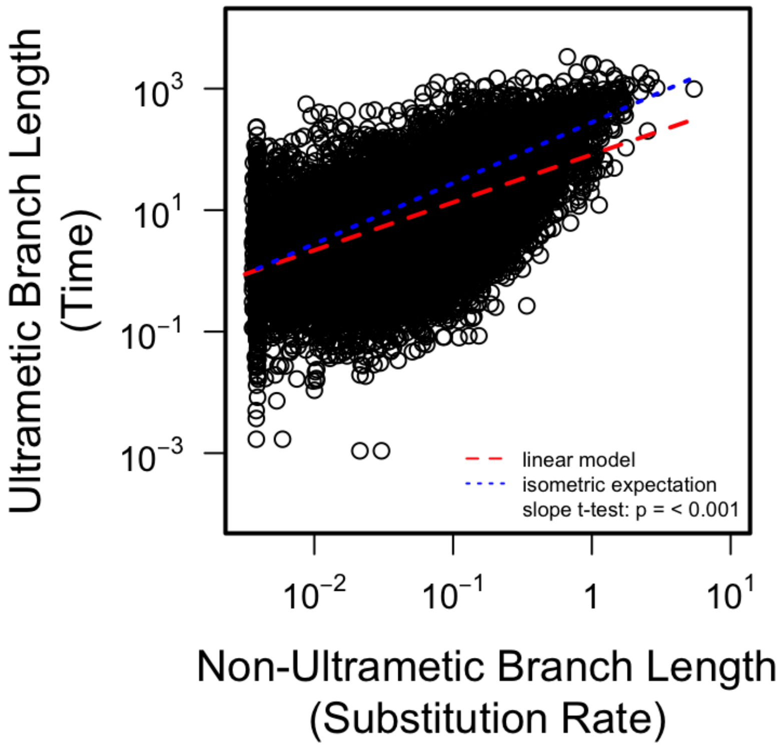
Branch Lenght Comparision

```r
png(filename="../figures/Branch_Comparision.png",
    width = 800, height = 800, res = 96*2)

reg <- livingTree.r2$edge.length
um <- livingTree.um$edge.length
```

```r
# Are nodes the same?
sum(livingTree.r2$node.label != livingTree.um$node.label)

zero.branch <- which(reg == 0 | um == 0)
reg.l <- log10(reg[-zero.branch]) + 2.5
um.l <- log10(um[-zero.branch])

mod1 <- lm(um.l ~ reg.l)
summary(mod1)
ttest(mod1, 2, 1)

par(mar = c(6,6,1,1) + 0.1, oma = c(0.5,0.5,0.5,0.5))
plot(reg.l, um.l,
     xlim = c(0, 3.5), ylim = c(-4, 4),
     axes=F, xlab = "", ylab = "")

# abline(mod1, from = -2.5, to = 1.5, lty = 2, lwd = 2, col = "red")
new <- data.frame(reg.l = seq(0, 3.2, 0.1))
lines(x = t(new), y = predict(mod1, new), lty = 2, lwd = 2, col = "red")
clip(min(reg.l), max(reg.l), min(um.l), max(um.l))
abline(mod1$coefficients[1], 1, lty = 3, lwd = 2, col = "blue", untf = T)

axis(side = 1, las = 1, at = c(seq(-3, 1, 1) + 2.5),
     labels = expression(10^-3, 10^-2, 10^-1, 1, 10^1))
axis(side = 2, las = 1, at = c(seq(-5,3,2)),
     labels = expression(10^-5, 10^-3, 10^-1, 10^1, 10^3))

legend("bottomright", legend = c("linear model", "isometric expectation",
                                 "slope t-test: p = < 0.001"),
       lty = c(2, 3, 0), col = c("red", "blue", "black"),
       bty = "n", cex = 0.6)

mtext("Non-Ultrametic Branch Length\n(Substitution Rate)", side = 1,line = 4, cex = 1.25)
mtext("Ultrametic Branch Length\n(Time)", side = 2, line = 3, cex = 1.25)

box(lwd = 2)

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```r
img <- readPNG("../figures/Branch_Comparision.png")
grid.raster(img)
```



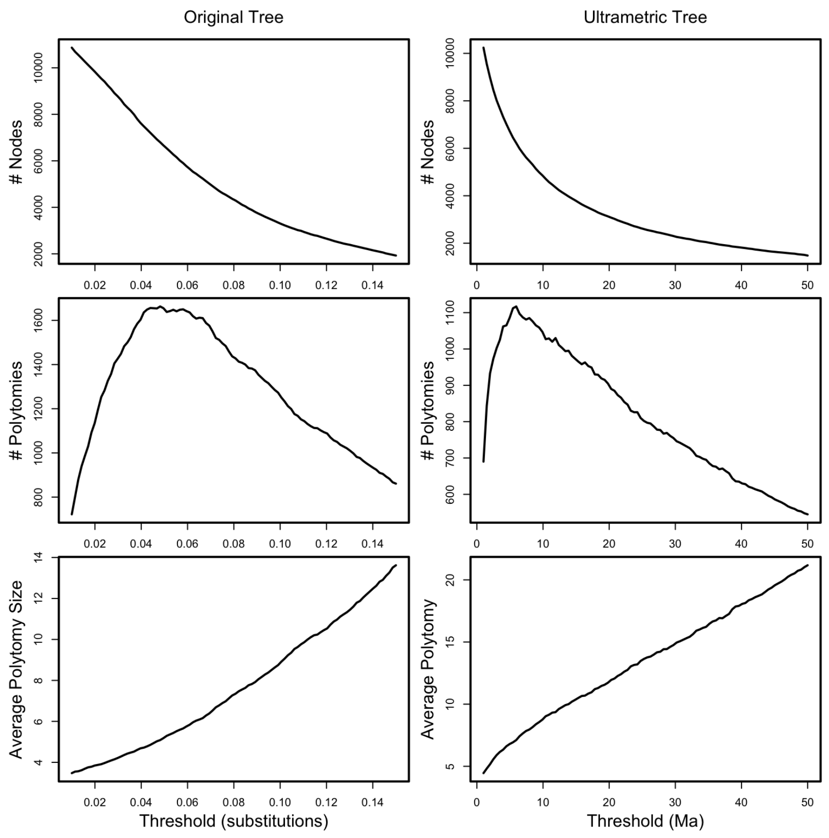
Phylogeny Topological Features

```
tree.thres <- data.frame(matrix(NA, 100, 4))
colnames(tree.thres) <- c("threshold", "nodes", "polytomies", "avg.polytomy")
tree.thres$threshold <- seq(0.01, 0.15, length.out = 100)

um.thres <- data.frame(matrix(NA, 100, 4))
colnames(um.thres) <- c("threshold", "nodes", "polytomies", "avg.polytomy")
um.thres$threshold <- seq(1, 50, length.out = 100)

phy1 <- livingTree.r2
phy2 <- livingTree.um

for (i in 1:dim(tree.thres)[1]){
  phy.grained <- di2multi2(phy1, tree.thres[i,1])
  polys <- table(table(phy.grained$edge[,1]))
  poly.size <- table(phy.grained$edge[,1])
  tree.thres[i, 2] <- phy.grained$Nnode
  tree.thres[i, 3] <- sum(polys[which(as.numeric(names(polys)) > 2)])
  tree.thres[i, 4] <- mean(poly.size[which(poly.size > 2)])
}

for (i in 1:dim(um.thres)[1]){
  phy.grained <- di2multi2(phy2, um.thres[i,1])
  polys <- table(table(phy.grained$edge[,1]))
  poly.size <- table(phy.grained$edge[,1])
  um.thres[i, 2] <- phy.grained$Nnode
  um.thres[i, 3] <- sum(polys[which(as.numeric(names(polys)) > 2)])
  um.thres[i, 4] <- mean(poly.size[which(poly.size > 2)])
}
```

```
png(filename="../figures/Tree_Topology.png",
    width = 1600, height = 1600, res = 96*2)

layout(matrix(1:6, 3))
par(mar = c(2,4,0.5,0.5), oma = c(3,1.5,3.5,1.5))
plot(tree.thres$nodes ~ tree.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("# Nodes", side = 2, line = 2.5)
mtext("Original Tree", side = 3, line = 1, outer = F)
box(lwd = 2)
plot(tree.thres$polytomies ~ tree.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("# Polytomies", side = 2, line = 2.5)
box(lwd = 2)
plot(tree.thres$avg.polytomy ~ tree.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("Threshold (substitutions)", side =1, line = 2.5, cex = 1)
mtext("Average Polytomy Size", side = 2, line = 2.5)
box(lwd = 2)

plot(um.thres$nodes ~ um.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("# Nodes", side = 2, line = 2.5)
mtext("Ultrametric Tree", side = 3, line = 1, outer = F)
box(lwd = 2)
plot(um.thres$polytomies ~ um.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("# Polytomies", side = 2, line = 2.5)
box(lwd = 2)
plot(um.thres$avg.polytomy ~ um.thres$threshold,
     xlab = "", ylab = "", type = "l", lwd = 2)
mtext("Threshold (Ma)", side =1, line = 2.5, cex = 1)
mtext("Average Polytomy", side = 2, line = 2.5)
box(lwd = 2)


# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/Tree_Topology.png")
grid.raster(img)
```



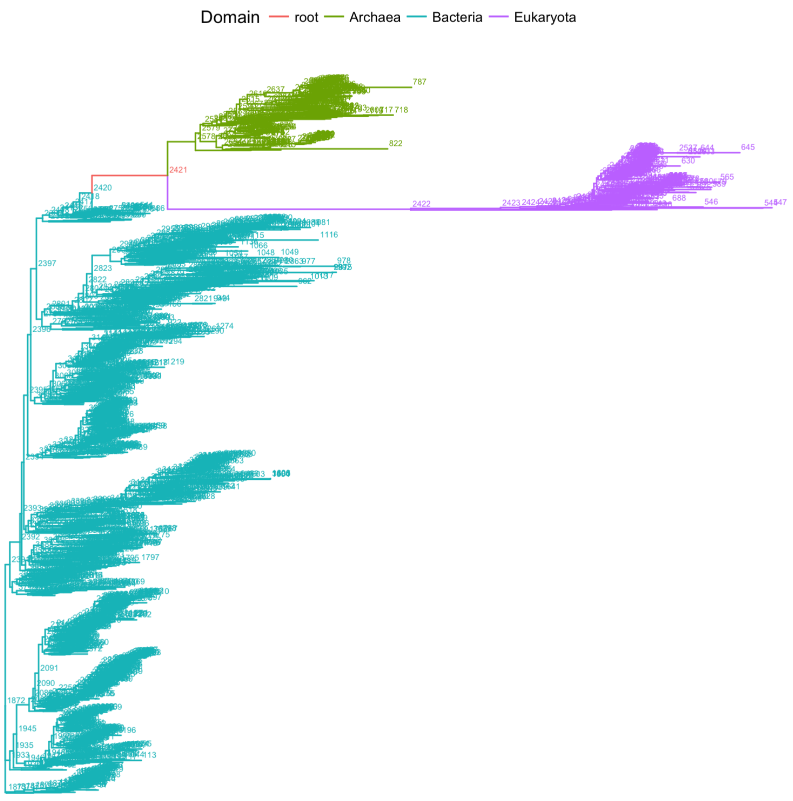
Repeat with Other trees (still working here)

NTL Exploratory Tree with Domain Grouping

```r
png(filename="../figures/NTL_explore.png",
    width = 1600, height = 1600, res = 96*2)

NTL.2 <- NTL
NTL.tax2 <- NTL.tax[na.omit(match(NTL.2$tip.label, NTL.tax$OTU)), ]
NTL.tax2[which(NTL.tax2$Domain == "unknown"), ]$Domain <- "Bacteria"
groupInfo <- split(NTL.2$tip.label, NTL.tax2$Domain)
newTree <- groupOTU(NTL.2, groupInfo, group_name = "Domain")
levels(attributes(newTree)$Domain) <- c("root", names(groupInfo))
newTree$node.label[which(attributes(newTree)$Domain == "root")]
ggtree(newTree, aes(color = Domain), layout="rectangular") +
  geom_text(aes(subset=!isTip, label = node), show.legend = FALSE,
            hjust=-0.1, vjust = -0.5, size = 2) +
  theme(legend.position="top", legend.key = element_rect(colour = NA))

# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/NTL_explore.png")
grid.raster(img)
```



Remove Eukaryota and Identify the NTL Node that seperates bacteria and archaea

```
png(filename="../figures/NTL_archaea.png",
    width = 1600, height = 1600, res = 96*2)
par(mar=c(1,1,1,3))
NTL.2 <- drop.tip(NTL, c(NTL.tax2$OTU[which(NTL.tax2$Domain == "Eukaryota")]))
NTL.tax3 <- NTL.tax2[-which(NTL.tax2$Domain == "Eukaryota"), ]
groupInfo <- split(NTL.2$tip.label, NTL.tax3$Domain)
newTree.2 <- groupOTU(NTL.2, groupInfo, group_name = "Domain")
cp <- ggtree(newTree.2, aes(color = Domain), layout="rectangular") %>% collapse(node=2420)
cp + geom_point2(aes(subset=(node == 18227)), size=3, shape=23,
                 colour="black", fill = "gray80") +
  theme(legend.position="top", legend.key = element_rect(colour = NA)) +
  geom_text2(aes(subset=(node == 18228), label = "Archaea"),
             show.legend = FALSE, hjust=-0.1, size = 3) +
  geom_text(aes(subset=!isTip, label = node), show.legend = FALSE,
            hjust=-0.1, vjust = -0.5, size = 2) +
  guides(colour = guide_legend(override.aes = list(shape = NA, label = NA)))
```

```
## Warning: Removed 351 rows containing missing values (geom_text).
```

```
# Close Plot Device
dev.off()
graphics.off()
```

Show Plot

```
img <- readPNG("../figures/NTL_archaea.png")
grid.raster(img)
```