

Generate LTDE phylogenetic tree

William R. Shoemaker, Jordan Bird, Stuart E. Jones, and Jay T. Lennon

24 February, 2018

1) Setup

A. Retrieve and Set Your Working Directory

```
rm(list = ls())
getwd()
#setwd("~/GitHub/LTDE/")
knitr::opts_knit$set(root.dir = '~/GitHub/LTDE/')
```

B. Load Packages

The `require()` function in R returns TRUE if the package was successfully loaded or FALSE if the package failed to load. This for loop loads each package and installs the package when `require()` returns FALSE.

```
# package.list <- c('ape', 'seqinr', 'phylobase', 'adephylo',
# 'geiger', 'picante', 'stats', 'RColorBrewer', #'caper',
# 'phylolm', 'pmc', 'ggplot2', 'tidyr', 'dplyr', 'phangorn',
# 'pander', 'phytools', 'psych')
library("ape")
library("seqinr")
```

2) Make the alignment

Run the file `persistence_707_712_721.sh`, which contains the following code.

```
#!/bin/bash
#PBS -k o
#PBS -l nodes=1:ppn=8,vmem=100gb,walltime=1:00:00
#PBS -M wrshoema@indiana.edu
#PBS -m abe
#PBS -j oe

module load gcc/4.9.2
module load boost/1.52.0
module load openmpi
module load mothur/1.38.1

cd /N/dc2/projects/Lennon_Sequences/LTDE_Tree

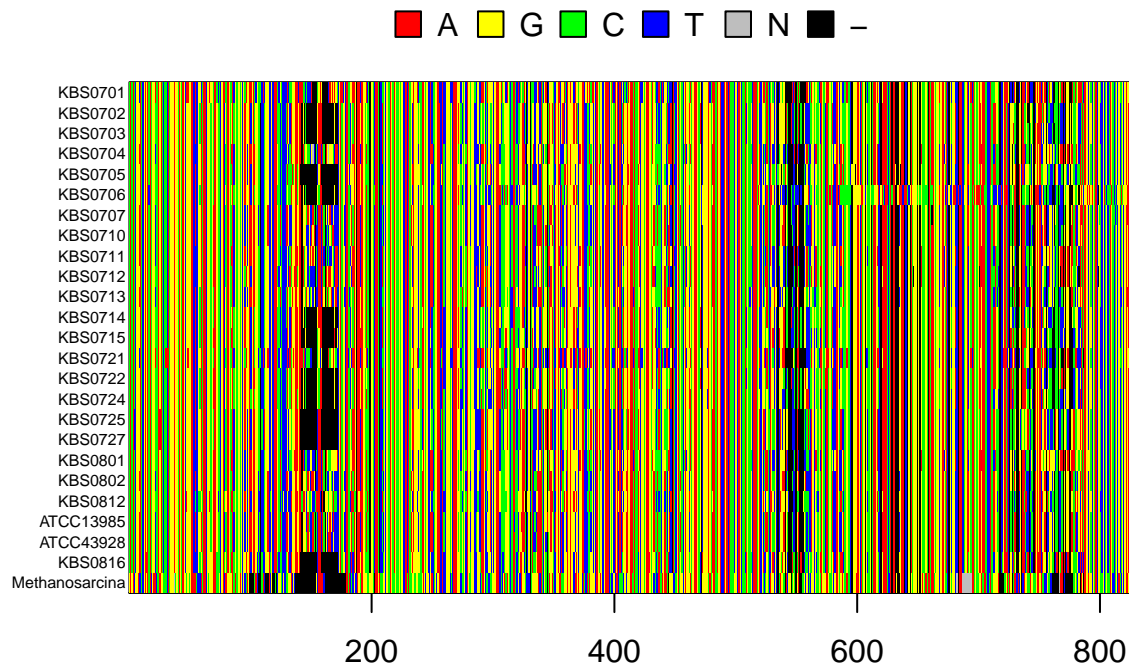
mothur persistence_707_712_721.batch
```

Which calls `persistence_707_712_721.batch`, a batch file containing the following instructions

```
align.seqs(fasta=persistence_707_712_721.fasta, reference=silva.v4.fasta, flip=T, processors=4)
align.seqs(fasta=persistence.fasta, reference=silva.bacteria.rdp.tax, flip=T, processors=4)
summary.seqs(fasta=persistence_707_712_721.align)
screen.seqs(fasta=persistence_707_712_721.align, minlength=200)
filter.seqs(fasta=persistence_707_712_721.good.align, vertical=T, trump=.)
```

3) View alignment

```
read.aln <- read.alignment(file = "data/align/persistence_707_712_721.good.filter.fasta", format = "fasta")
p.DNABin <- as.DNABin(read.aln)
# Identify Base Pair Region of 16S rRNA Gene to Visualize (adjust range)
>window.M <- p.DNABin.M[, 1:200]
# Command to Visualize Sequence Alignment {ape}
image.DNABin(p.DNABin, cex.lab = 0.50)
```



4) Build the tree

We then build the filtered FASTA formatted alignment file using RAxML

```
raxmlHPC-PTHREADS -T 4 -f a -m GTRGAMMA -p 12345 -x 12345 -o Methanosarcina -# autoMRE \
-s ~/GitHub/LTDE/data/align/persistence_707_712_721.good.filter.fasta \
-n T20 -w ~/GitHub/LTDE/data/tree
```

```
# -T = number of threads
# -f = specifies bootstrapping algorithm with ML generating tree at same time
# -m = substitution model, generalized time reversible gamma
# -p = starts tree randomly
# -x = starts tree randomly
# -o = outgroup (name after fasta entry)
# -# = determines number of bootstrap replicates
```

```
# -s = aligned fasta file to be analyzed
# -n = name of output file
```

5) Visualize the tree

```
# Read tree
ml.tree <- read.tree("data/tree/RAXML_bipartitionsBranchLabels.T20")

# Identify Outgroup Sequence
outgroup <- match("Methanosarcina", ml.tree$tip.label)

# Root the Tree {ape}
ml.rooted <- root(ml.tree, outgroup, resolve.root = TRUE)

# Load phylo taxonomy data
tax <- read.table("data/traits/persistence.phylo.txt", sep = "\t", header = TRUE)
#rownames(tax) <- tax$Code
#tax.2 <- tax[ml.tree$tip.label, ]
#tax.name <- paste(tax.2$Code, tax.2$Genus)
#tax.name[15] <- "DSM2834 Methanosarcina"

# Plot the Rooted Tree{ape}
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(ml.rooted, main = "RAXML Tree",
           "phylogram", use.edge.length = FALSE, direction = "right",
           cex = 0.6, label.offset = 1, show.tip.label = FALSE, x.lim = 30)

tiplabels(ml.rooted$tip.label, adj = c(0,0.5), cex = 0.5, frame = "none",
           pch = NULL, thermo = NULL, pie = NULL, piecol = NULL,
           col = NULL, bg = NULL)

nodelabels(ml.rooted$node.label, font = 2, bg = "white", frame = "r", cex = 0.5)
add.scale.bar(0.4, 0.4, cex = 0.7)
```

RAXML Tree

