```
#install.packages("ape")

# Load rhmmer for parsing HMMER results.
library(rhmmer)

# Import pheatmap library for making nice heatmaps.
library("pheatmap")
library("RColorBrewer")

# Import seqinr for editing fasta files.
library(seqinr)

# Import stringr for matching strings.
library(stringr)

# Import ape for dealing with phylogenetic trees.
library(ape)

# Remove all objects.
rm(list = ls())

# Remove all objects.
rm(list = ls())

# For getting all the google colours.
load("/home/kim_lab_kvm/Documents/Species_Phylogeny/google_colours.RData"
)

# Set the working directory.
setwd("/home/kim_lab_kvm/Documents/Species_Phylogeny/bacterial/")

# Set location of programs to run.
trimal <- "/home/kim_lab_kvm/miniconda3/envs/py3/bin/trimal"
mafft <- "/usr/local/bin/mafft"
fasttree <- "/home/kim_lab_kvm/miniconda3/envs/py3/bin/fasttree"

# Experiment - always use refseq rather than genbank data. The latter
contains unfinished genomes, which are more complicated to analyse.
# Read in experiment from command line arguments
args = commandArgs(trailingOnly=TRUE)
if (length(args)==0) {
  stop("At least one argument must be supplied (experiment).n",
call.=FALSE)
} else if (length(args)==1) {
  experiment <- args[1]
}

# Get the getHousekeeping workspace.
load(paste("data/", experiment, "/housekeeping_fasta_from_hmm.RData", sep
= ""))

# Remove old folders and create new folder for
file.fasta.translated.cdsein sequences.
```

```
command <- paste("rm -r data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/ 2> /dev/null",
sep = "")
system(command, intern = TRUE)
command <- paste("mkdir -p data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/", sep = "")
system(command, intern = TRUE)

# Remove old folders and create new folder for
file.fasta.translated.cdsein sequences.
command <- paste("rm -r data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_concatenated/ 2>
/dev/null", sep = "")
system(command, intern = TRUE)
command <- paste("mkdir -p data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_concatenated/", sep = "")
system(command, intern = TRUE)

# Vector to hold concatenated gene sequences.
cds.concatenated <- vector(mode = "character", length = num.genomes)
translated.cds.concatenated <- vector(mode = "character", length =
num.genomes)

for(j in 1:num.bac120.hmm){

  # Make directory for all sequences of given hmm.
  command <- paste("mkdir -p data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/",
data.bac120.hmm.info$accession[j], sep = "")
  system(command, intern = FALSE)

  # Add the cds sequences to single fasta file.
  file.fasta.cds <- paste("data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/",
data.bac120.hmm.info$accession[j], "/",
data.bac120.hmm.info$accession[j], ".fna", sep = "")
  command <- paste("cat data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds/",
data.bac120.hmm.info$accession[j], "/*.fna > ", file.fasta.cds, sep = "")
  system(command, intern = FALSE)

  # Add the translated cds sequences to single fasta file.
  file.fasta.translated.cds <- paste("data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/",
data.bac120.hmm.info$accession[j], "/",
data.bac120.hmm.info$accession[j], ".faa", sep = "")
  command <- paste("cat data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds/",
data.bac120.hmm.info$accession[j],"/*.faa > ", file.fasta.translated.cds,
sep = "")
  system(command, intern = FALSE)

  # Get the fasta data.
```

```
  fasta.cds <- read.fasta(file = file.fasta.cds, seqtype = "DNA",
as.string = TRUE, forceDNAtolower = FALSE)
  fasta.translated.cds <- read.fasta(file = file.fasta.translated.cds,
seqtype = "AA", as.string = TRUE, forceDNAtolower = FALSE)

  # Create a data frame to store cds information.
  data.cds.info <- data.frame(matrix(NA, nrow = num.genomes, ncol = 0))
  data.cds.info$name <- getName(fasta.cds)
  data.cds.info$gene <- str_match(getAnnot(fasta.cds), "gene=(.*?)\\]")[
, 2]
  data.cds.info$locus_tag <- str_match(getAnnot(fasta.cds),
"locus_tag=(.*?)\\]")[ , 2]
  data.cds.info$db_xref <- str_match(getAnnot(fasta.cds),
"db_xref=(.*?)\\]")[ , 2]
  data.cds.info$protein <- str_match(getAnnot(fasta.cds),
"protein=(.*?)\\]")[ , 2]
  data.cds.info$protein_id <- str_match(getAnnot(fasta.cds),
"protein_id=(.*?)\\]")[ , 2]
  data.cds.info$location <- str_match(getAnnot(fasta.cds),
"location=(.*?)\\]")[ , 2]
  data.cds.info$gbkey <- str_match(getAnnot(fasta.cds),
"gbkey=(.*?)\\]")[ , 2]
  data.cds.info$ncbi.accession <- gsub("^(.*?)_cds_", "",
data.cds.info$name)
  data.cds.info$ncbi.accession <- gsub("_[^_]+$", "",
data.cds.info$ncbi.accession)

  # Create fasta header.
  data.cds.info$fasta.header <- paste(data.cds.info$ncbi.accession, " ",
data.assembly.info$species.strain.isolate, " ", data.cds.info$gene, " ",
data.cds.info$protein, sep = "")

  # Edit the fasta headers and remove problematic characters for
downstream newick tree format.
  data.cds.info$fasta.header <- gsub(",", " ",
data.cds.info$fasta.header)
  data.cds.info$fasta.header <- gsub(":", " ",
data.cds.info$fasta.header)
  data.cds.info$fasta.header <- gsub(";", " ",
data.cds.info$fasta.header)
  data.cds.info$fasta.header <- gsub("\\(", "\\[",
data.cds.info$fasta.header)
  data.cds.info$fasta.header <- gsub("\\)", "\\]",
data.cds.info$fasta.header)

  # Write to tsv file.
  file.cds.info <- paste("data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_combined/",
data.bac120.hmm.info$accession[j], "/",
data.bac120.hmm.info$accession[j], ".tsv", sep = "")
  write.table(data.cds.info, file = file.cds.info, quote = FALSE, sep =
"\t", col.names = NA)

  # Run mafft alignment on file.fasta.translated.cdseins.
```

```
  file.fasta.translated.cds.aln <- paste(file.fasta.translated.cds,
".aln", sep = "")
  command <- paste(mafft, " --maxiterate 1000 --thread 16 --globalpair ",
file.fasta.translated.cds, " > ", file.fasta.translated.cds.aln, sep =
"")
  system(command, intern = TRUE)

  # Get the fasta data.
  fasta.translated.cds.aln <- read.fasta(file =
file.fasta.translated.cds.aln, seqtype = "AA", as.string = TRUE,
forceDNAtolower = FALSE)

  # Run trimal to remove gappy regions in file.fasta.translated.cdsein
alignment.
  file.fasta.translated.cds.aln.trimal <-
paste(file.fasta.translated.cds.aln, ".trimal", sep = "")
  command <- paste(trimal, " -in ", file.fasta.translated.cds.aln, " -out
", file.fasta.translated.cds.aln.trimal, " -gapthreshold 0.9 -fasta")
  system(command)

  # Read in the fasta data.
  fasta.translated.cds.aln.trimal <- read.fasta(file =
file.fasta.translated.cds.aln.trimal, seqtype = "AA", as.string = TRUE,
forceDNAtolower = FALSE)

  # Write the new file.fasta.translated.cds fasta (need to do this
because alignment process changes the headers).
  file.fasta.translated.cds.aln.trimal.renamed <-
paste(file.fasta.translated.cds.aln.trimal, ".renamed", sep = "")
  write.fasta(sequences = getSequence(fasta.translated.cds.aln.trimal),
names = data.cds.info$fasta.header, file.out =
file.fasta.translated.cds.aln.trimal.renamed)

  # Add file.fasta.translated.cds to concatenated data.
  translated.cds.concatenated <- paste(translated.cds.concatenated,
unlist(getSequence(fasta.translated.cds.aln.trimal, as.string = TRUE)),
sep = "")

  # Run fasttree to create an individual gene tree.
  # file.newick.translated.cds <-
paste(file.fasta.translated.cds.aln.trimal.renamed, ".newick", sep = "")
  # command <- paste(fasttree, " -wag -gamma -quote < ",
file.fasta.translated.cds.aln.trimal.renamed, " > ",
file.newick.translated.cds, sep = "")
  # system(command)

}

# Create fasta header.
translated.cds.concatenated.fasta.header <-
paste(data.assembly.info$refseq.genome.id)

# Edit the fasta headers and remove problematic characters for downstream
newick tree format.
```

```
translated.cds.concatenated.fasta.header <- gsub(",", " ",
translated.cds.concatenated.fasta.header)
translated.cds.concatenated.fasta.header <- gsub(":", " ",
translated.cds.concatenated.fasta.header)
translated.cds.concatenated.fasta.header <- gsub(";", " ",
translated.cds.concatenated.fasta.header)
translated.cds.concatenated.fasta.header <- gsub("\\(", "\\[",
translated.cds.concatenated.fasta.header)
translated.cds.concatenated.fasta.header <- gsub("\\)", "\\]",
translated.cds.concatenated.fasta.header)

# Write out the fasta data.
file.fasta.translated.cds.concatenated <- paste("data/", experiment,
"/hmmsearch_bac120_output_from_translated_cds_concatenated/bac120.faa.aln
.trimal.renamed", sep = "")
write.fasta(sequences = as.list(translated.cds.concatenated), names =
translated.cds.concatenated.fasta.header, file.out =
file.fasta.translated.cds.concatenated, as.string = TRUE)

# Run fasttree to create a tree.
#file.newick.translated.cds.concatenated <-
paste(file.fasta.translated.cds.concatenated, ".newick", sep = "")
#command <- paste(fasttree, " -wag -gamma -quote < ",
file.fasta.translated.cds.concatenated, " > ",
file.newick.translated.cds.concatenated, sep = "")
#system(command = command, intern = FALSE)

# Read in tree.
#newick.translated.cds.concatenated <-
read.tree(file.newick.translated.cds.concatenated)

# Change headers if desired.
# newick.translated.cds.concatenated$tip.label <-
translated.cds.concatenated.fasta.header
# write.tree(newick.translated.cds.concatenated, file =
file.newick.translated.cds.concatenated)

# Save the workspace so it can be loaded from start without having to
repeat all these steps.
#file.rdata <- paste("data/", experiment,
"/housekeeping_tree_from_fasta.RData", sep = "")
#save(list = ls(all.names = TRUE), file = file.rdata)
```