Test LTDE phylogenetic signal

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

24 February, 2018

Overview

1) Set working directory and load packages

```
# setup
## Retrieve and Set Your Working Directory
rm(list = ls())
getwd()
```

```
## [1] "/Users/WRShoemaker/GitHub/LTDE/R"
```

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

library('pracma')
library('ape')
library('phylolm')
library('phytools')
library('lmtest')
library('lme4')
require("png")
require("grid")
require("vegan")
```

2) Load population data

```
df <- read.table("data/demography/weibull_results.csv",
                 header = TRUE, sep = ",", row.names = 1)
# rename mis-named reps
df$rep[df$strain == "KBS0711W" & df$rep == "1"] <- 5
df$rep[df$strain == "KBS0711W" & df$rep == "2"] <- 6
df$rep[df$strain == "KBS0711W" & df$rep == "3"] <- 7
df$rep[df$strain == "KBS0711W" & df$rep == "4"] <- 8
# rename mis-named strain
df$strain[df$strain == "KBS0711W"] <- "KBS0711"
```

3) Calculate Weibull survival function half-life and death rate

Because no energy is entering our experimental system, it is fair to assume that the rate of death (d) is much higher than the rate of birth (b). That is, $d >> b$. Based off of this assumption, we chose to analyze the death rates of our experimental population using survival analysis. Survival analysis primarily focuses on estimating the probability that the time of death of a system is later than some specified time t. This probability is characterized by the survival function:

$$S(t) = \Pr(T > t)$$

where T is random variable representing the time of death of the system, t is some period of time, and Pr stands for probability. Because $S(t)$ is defined a probility, its value starts at 0 and eventually reaches 1, meaning that $S(t)$ can be defined as the complement of a probability density function ($F(t)$) of our choice:

$$S(t) = \Pr(T > t) = 1 - \Pr(T \leq t) = 1 - F(t)$$

Because our death curves display a non-linear decay on a log scale, we chose to use the Weibull distribution. The survival function of the Weibull distribution is:

$$S(t) = e^{-(t/\beta)^{\alpha}}$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter. If $\alpha = 1$ then the failure rate of the system is constant and $S(t)$ reduces to an exponential distribution. However, a value of $\alpha < 1$ indicates that the failure rate decreases over time, the opposite being true for $\alpha > 1$.

Using this distribution we calculated the average length of time until an individual dies ($\bar{T_d}$), often referred in the survival analysis literature as Mean Time To Failure (MTTF). For a given $S(t)$, $\bar{T_d}$ is just the area under the curve of the survival function

$$\bar{T_d} = \int_0^{\infty} S(t)dt$$

This function can be converted into a simpler form as follows. First, we add a term to the right hand side of the equation that is equal to zero

$$\bar{T_d} = [-tS(t)] \,|_0^{\infty} + \int_0^{\infty} S(t)dt$$

where the right hand side of the equation can now be interpreted as the integral of the derivative of the survival function multiplied by time

$$\bar{T_d} = -\int_0^{\infty} t\, d[S(t)]$$

Using our definition of the survival function, we can replace $S(t)$ with its complement

$$\bar{T_d} = -\int_0^{\infty} t\, d[1 - F(t)]$$

$$= \int_0^{\infty} t\, dF(t)$$

since the cumulative distribution function is the integral of the probability density function, we get

$$\bar{T_d} = \int_0^{\infty} t\, f(t)dt$$

Which is just the expected value ($E[t]$) of the Weibull distribution

$$\bar{T}_d = \beta\,\Gamma\left(1 + 1/\alpha\right)$$

where $\Gamma(x)$ is the gamma function.

The average length of time before half the population dies ($t_{1/2}$) for the Weibull survival function is:

$$\beta * (-\ln(0.5))^{\alpha^{-1}}$$

```
# get MTTF
weibull_mean <- function(alpha, beta){
  return(beta * gamma(1 + (1/alpha)))
}
df$weibull_mean <- weibull_mean(df$alpha, df$beta)
# get half-life for Weibull survival function
weibull_half_life <- function(alpha, beta){
  return(beta *  ((-log(0.5)) ^ (1/alpha))  )
}
df$weibull_half_life <- weibull_half_life(df$alpha, df$beta)
# setting alpha = 1, determne MTTF if death rate doesn't change with time
df$exp_mean <-  df$beta
# log base 10 transform the parameters
df$weibull_mean_log10 <- log10(df$weibull_mean)
df$exp_mean_log10 <- log10(df$exp_mean)
```

4) Perform linear and quadratic regression

do for death rate when alpha = 1 (exponential decay)

```
# try with no bacillus
#df.noB <- df[!df$strain=="KBS0812",]
df.noB <- df
linear.re.inter <- lmer(weibull_mean_log10 ~ exp_mean_log10 + (1 | strain), data=df.noB)
quad.re.inter <- lmer(weibull_mean_log10 ~ exp_mean_log10
                      + I(exp_mean_log10^2) + (1 | strain), data=df.noB)
# significant lrt, keep random slopes
lrtest(quad.re.inter, linear.re.inter)
```

```
## Likelihood ratio test
##
## Model 1: weibull_mean_log10 ~ exp_mean_log10 + I(exp_mean_log10^2) + (1 |
##     strain)
## Model 2: weibull_mean_log10 ~ exp_mean_log10 + (1 | strain)
##   #Df   LogLik Df  Chisq Pr(>Chisq)
## 1   5  -45.098
## 2   4 -133.077 -1 175.96  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
a <- fixef(quad.re.inter)
linear.re.inter.ci <- confint(linear.re.inter)
quad.re.inter.ci <-  confint(quad.re.inter)

# make plot
```

```r
x_y <- df.noB[c("strain","exp_mean_log10","weibull_mean_log10")]
x_y.mean <- aggregate(.~strain, data=x_y, mean)
colnames(x_y.mean)[colnames(x_y.mean)=="exp_mean_log10"] <- "exp_mean_log10_mean"
colnames(x_y.mean)[colnames(x_y.mean)=="weibull_mean_log10"] <- "weibull_mean_log10_mean"
x_y.sd <- aggregate(.~strain, data=x_y, sd)
colnames(x_y.sd)[colnames(x_y.sd)=="exp_mean_log10"] <- "exp_mean_log10_sd"
colnames(x_y.sd)[colnames(x_y.sd)=="weibull_mean_log10"] <- "weibull_mean_log10_sd"
x_y.len <- aggregate(.~strain, data=x_y, length)
colnames(x_y.len)[colnames(x_y.len)=="exp_mean_log10"] <- "n"
x_y.len<-x_y.len[ , !(names(x_y.len) %in% c('weibull_mean_log10'))]

merge.1 <- merge(x_y.mean, x_y.sd, by = "strain")
merge.final <- merge(merge.1, x_y.len, by = "strain")
merge.final$exp_error <- qt(0.975,df=merge.final$n-1) *
  merge.final$exp_mean_log10_sd/sqrt(merge.final$n)
merge.final$weib_error <- qt(0.975,df=merge.final$n-1) *
  merge.final$weibull_mean_log10_sd/sqrt(merge.final$n)
merge.final$exp_CI.L <- merge.final$exp_mean_log10_mean - merge.final$exp_error
merge.final$exp_CI.U <- merge.final$exp_mean_log10_mean + merge.final$exp_error
merge.final$weib_CI.L <- merge.final$weibull_mean_log10_mean - merge.final$weib_error
merge.final$weib_CI.U <- merge.final$weibull_mean_log10_mean + merge.final$weib_error

png(filename = paste(c("figs/exp_weib_T_d.png"), collapse = ''),
    width = 1200, height = 1200, res = 96*2)

par(mar = c(4, 4, 1, 1))
plot(merge.final$exp_mean_log10_mean,
     merge.final$weibull_mean_log10_mean, ylab = "", xlab = "")
#arrows(x0 = merge.final$exp_mean_log10_mean, y0 = merge.final$weibull_mean_log10_mean,
#       y1 = merge.final$weib_CI.L, angle = 90, length = 0.1, lwd = 1)
#arrows(x0 = merge.final$exp_mean_log10_mean, y0 = merge.final$weibull_mean_log10_mean,
#       y1 = merge.final$weib_CI.U, angle = 90, length = 0.1, lwd = 1)

#arrows(x0 = merge.final$exp_mean_log10_mean, x1 = merge.final$exp_CI.L,
#       y0 = merge.final$weibull_mean_log10_mean, angle = 90, length = 0.1, lwd = 1)
#arrows(x0 = merge.final$exp_mean_log10_mean, x1 = merge.final$exp_CI.U,
#       y0 = merge.final$weibull_mean_log10_mean, angle = 90, length = 0.1, lwd = 1)

abline(0, 1, lw = 2)
#abline(fixef(linear.re.inter),col="red")
# plot linear and quadratic fits +/- 95% CIs
minMax <- range(merge.final$exp_mean_log10_mean)
xVals <- seq(minMax[1], minMax[2], len = 100)
yVals.lin <- a['(Intercept)'] + xVals * a['exp_mean_log10']
lines(xVals, yVals.lin, col = 'red', lw = 2)
lines(xVals, yVals.lin-linear.re.inter.ci['exp_mean_log10','2.5 %'],
      col = 'red', lty = 'dashed')
lines(xVals, yVals.lin+linear.re.inter.ci['exp_mean_log10','97.5 %'],
      col = 'red', lty = 'dashed')

yVals.quad <- a['(Intercept)'] + xVals * a['exp_mean_log10'] +
  (xVals^2) * a['I(exp_mean_log10^2)']
lines(xVals, yVals.quad, col = 'blue', lw = 2)
```

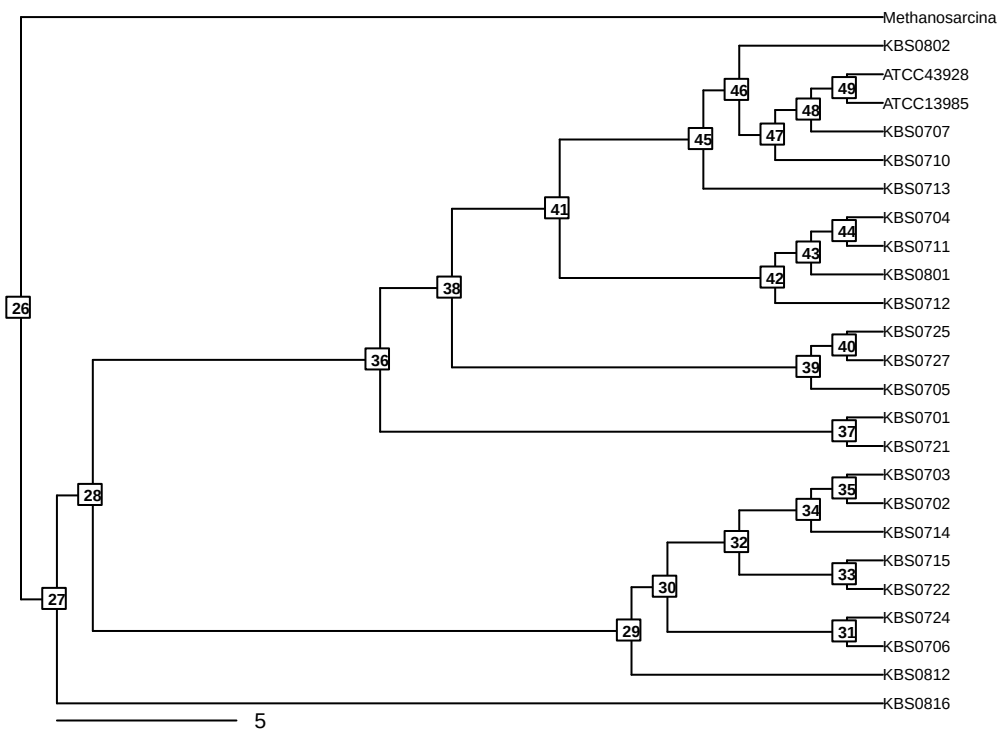
```
lines(xVals, yVals.quad-quad.re.inter.ci['exp_mean_log10','2.5 %'],
      col = 'blue', lty = 'dashed')
lines(xVals, yVals.quad+quad.re.inter.ci['exp_mean_log10','97.5 %'],
      col = 'blue', lty = 'dashed')

mtext(expression('Exponential average death rate, log 10'), side = 1,
      outer = FALSE, cex = 1, line = 2.6, adj = 0.7)

mtext(expression('Weibull average death rate, log 10'), side = 2,
      outer = FALSE, cex = 1, line = 2, adj = 0.5)

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

```
## pdf
##   2
```

```
graphics.off()

# Show Plot
img <- readPNG(paste(c("figs/exp_weib_T_d.png"), collapse = ''))
grid.raster(img)
```

5) Load phylogenetic data

```
# Load ML tree
ml.tree <- read.tree("data/tree/RAxML_bipartitionsBranchLabels.T20")
# Define the outgroup
outgroup <- match("Methanosarcina", ml.tree$tip.label)
# Create a rooted tree {ape} using the outgroup
ml.rooted <- root(ml.tree, outgroup, resolve.root = TRUE)
# Plot the Rooted Tree{ape}
par(mar = c(1,1,2,1) + 0.1)
plot.phylo(ml.rooted, main = "raxml",
           "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(1, 0.4, cex = 0.7)
```



6) Run phylogenetic corrected regression

```
# Keep rooted but drop outgroup branch
ml.rooted <- drop.tip(ml.rooted, c("Methanosarcina"))
is.ultrametric(ml.rooted)
```

```
## [1] FALSE
```

```
ml.rooted.um  <- chronos(ml.rooted)
```

```
##
## Setting initial dates...
## Fitting in progress... get a first set of estimates
##          Penalised log-lik = -8.321145
## Optimising rates... dates... -8.321145
##
## Done.
```

```
is.ultrametric(ml.rooted.um)
```

```
## [1] TRUE
```

```
# pmc is having trouble converting the chronos object to a phy type object
# just save the tree and re-load it
write.tree(ml.rooted.um, file = "data/tree/test.txt")
re.ml.rooted.um <- read.tree("data/tree/test.txt")


rownames(merge.final) <- merge.final$strain
```

```
#merge.final[,c("weibull_mean_log10_mean")]
#weib <- as.matrix(merge.final$weibull_mean_log10_mean)
#rownames(weib) <-  merge.final$strain

re.ml.rooted.um.prunned <- drop.tip(re.ml.rooted.um,
                                    re.ml.rooted.um$tip.label[na.omit(match(c('KBS0816', 'KBS0704'),
                                    re.ml.rooted.um$tip.label))])


# Run a phylogeny-corrected regression with no bootstrap replicates
#fit.phy <- phylolm(weibull_mean_log10_mean ~ exp_mean_log10_mean, data = merge.final,
#                   re.ml.rooted.um.prunned, model = 'lambda', boot = 0)
```

6) Determine if metabolic pathways explain death rate

```
#df.met <- read.table("data/metab_paths/module_by_taxon.txt",
#                 header = TRUE, sep = "\t", row.names = 1)
# remove rows with all ones
#df.met<- t(df.met[apply(df.met[,-1], 1, function(x) !all(x==1)),])

#df.met.db <- vegdist(df.met, method = "bray", upper = TRUE, diag = TRUE, binary=TRUE)
#df.met.pcoa <- cmdscale(df.met.db, eig = TRUE, k = 2)


#df.met.pca <- phyl.pca(re.ml.rooted.um, df.met, method = "lambda")
# df.met.pca$S
# merge rownames
#df.met.merge <- merge(merge.final, df.met.pcoa$point, by="row.names")

#plot(df.met.pcoa$points[,1],
#     df.met.pcoa$points[,2], ylab = "", xlab = "")

#plot(df.met.merge$V1,
#     df.met.merge$weibull_mean_log10_mean, ylab = "", xlab = "")

# no real relationship between first pcoa axis and average time to extinction
```

7) Load trait data

```
traits <- read.table("data/traits/persistence.phylo.txt", sep = "\t", header = TRUE)
rownames(traits) <- traits$Code
traits <- traits[re.ml.rooted.um$tip.label, ]
# doubling time = log(2) / umax
# what units are umax in?
traits$T_birth <- log(2) / traits$umax

# calculate average doubling time (i.e., average birth rate)
```