

Test LTDE phylogenetic signal

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

22 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('pmc')
library('lmtest')
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
library('lme4')
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:pracma':
```

```
##
```

```
##      expm, lu, tril, triu
```

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, $d \gg b$. 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 probability, its value starts at 1 and eventually reaches 0, 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 a population goes extinct, often referred in the survival analysis literature as Mean Time To Failure (MTTF). For a given $S(t)$, MTTF is just the area under the curve of the survival function

$$\text{MTTF} = \int_0^\infty S(t) dt$$

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

$$\text{MTTF} = [-tS(t)] \Big|_0^\infty + \int_0^\infty S(t) dt$$

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

$$\text{MTTF} = - \int_0^\infty t d[S(t)]$$

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

$$\text{MTTF} = - \int_0^\infty t d[1 - F(t)]$$

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

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

$$\text{MTTF} = \int_0^{\infty} t f(t) dt$$

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

$$\text{MTTF} = \beta \Gamma(1 + 1/\alpha)$$

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, determine 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) Load phylogenetic data

```
# Load ML tree
ml.tree <- read.tree("data/tree/RAxML_bipartitionsBranchLabels.T20")

# Define the outgroup
outgroup <- match("Methanosarcina", ml.tree$tip.label)
# Create a rooted tree {ape} using the outgroup
ml.rooted <- root(ml.tree, outgroup, resolve.root = TRUE)
# Keep rooted but drop outgroup branch
# Also drop other taxa lacking trait data
#ml.rooted <- drop.tip(ml.rooted, c("Methanosarcina", "KBS0816", "KBS0704", "KBS0727B"))
ml.rooted <- drop.tip(ml.rooted, c("Methanosarcina"))
ml.rooted$tip.label['KBS0725B'] <- "KBS0725"
ml.rooted$tip.label['KBS0727B'] <- "KBS0727"
```

4) Test for phylogenetic signal

```
#is.ultrametric(ml.rooted)
#ml.rooted.um <- chronos(nj.rooted)
```

5) 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)
##      #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
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$exp_mean_log10_sd/sqrt(merge.final$exp_mean_log10_mean)
merge.final$weib_error <- qt(0.975, df=merge.final$n-1) * merge.final$weibull_mean_log10_sd/sqrt(merge.final$weibull_mean_log10_mean)
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
```

```

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$weibull_mean_log10_mean,
arrows(x0 = merge.final$exp_mean_log10_mean, y0 = merge.final$weibull_mean_log10_mean, y1 = merge.final$weibull_mean_log10_mean,

arrows(x0 = merge.final$exp_mean_log10_mean, x1 = merge.final$exp_CI.L, y0 = merge.final$weibull_mean_log10_mean,
arrows(x0 = merge.final$exp_mean_log10_mean, x1 = merge.final$exp_CI.U, y0 = merge.final$weibull_mean_log10_mean,

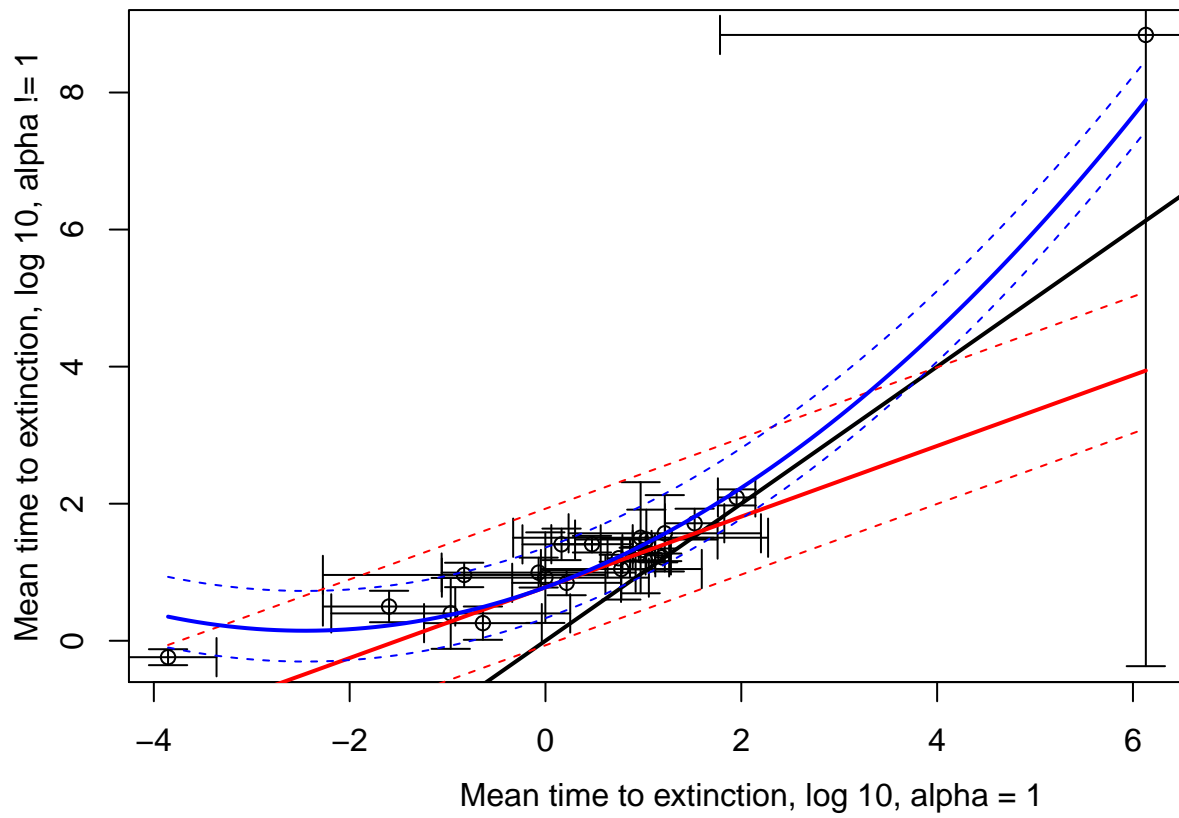
abline(0, 1, lw = 2)
#abline(fixef(linear.re.inter), col="red")
# plot linear and quadratic fits +/- 95% CIs
minMax <- range(merge.final$exp_mean_log10_mean)
xVals <- seq(minMax[1], minMax[2], len = 100)
yVals.lin <- a['(Intercept)'] + xVals * a['exp_mean_log10']
lines(xVals, yVals.lin, col = 'red', 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('Mean time to extinction, log 10, alpha = 1'), side = 1,
      outer = FALSE, cex = 1, line = 2.6, adj = 0.7)

mtext(expression('Mean time to extinction, log 10, alpha != 1'), side = 2,
      outer = FALSE, cex = 1, line = 2, adj = 0.5)

```



6) Merge trait data....

```
# calculate average doubling time (i.e., average birth rate)
#  $\ln(2) / u_{max}$ 
```