

# Biofilm Traits

*Jay T. Lennon*

*08 July, 2015*

Project looking at biofilm production of *Pseudomonas aeruginosa* in soil environment

```
rm(list=ls())  
getwd()
```

```
## [1] "/Users/lennonj/GitHub/Biofilms"
```

```
setwd("/Users/lennonj/GitHub/Biofilms")
```

```
require("png")
```

```
## Loading required package: png
```

## SUPPLEMENTARY FIGURE 1: BIOFILM-LAG TIME TRADEOFF

Read in trait data from Lennon et al. (2012)

```
traits <- read.table("data/Biofilm_Lennon_2012.txt", sep="\t", header=TRUE)
```

Remove rows with NAs

```
traits <- traits[complete.cases(traits),]
```

Plot biofilm and lag time

```
png(filename="/Users/lennonj/GitHub/Biofilms/figures/SupplementalFigure1.png",  
     width = 1200, height = 1200, res = 96*2)  
  
par(mar = c(5, 6, 4, 2))  
plot(traits$lag_time, traits$biofilm, xlab = "Lag time (hr)",  
     ylab = expression('Biofilm Production (a'[550]*')'), pch = 22,  
     cex = 3, col = "black", cex.lab = 2, las = 1, lwd = 2,  
     yaxt = "n", xaxt = "n")  
box(lwd = 2)  
  
axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,  
     labels = c("0.0", "1.0", "2.0", "3.0"), at = c(0, 1.0, 2.0, 3.0))  
  
axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,  
     at = c(0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5), labels = F)  
  
axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1,  
     labels = c("0", "20", "40", "60", "80"), at = c(0, 20, 40, 60, 80))
```

```
axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
      at = c(0, 20, 40, 60, 80), labels = F)

# Add p-value for t-test
mtext(expression(~italic(rho)~"= 0.38"), line = -2, cex = 1, at = 70)
mtext(expression(~italic("P")~"= 0.015"), line = -3, cex = 1, at = 70)

#mtext(expression(paste(rho, "= -0.38"), line = -10, cex = 1, at = 70))

dev.off() # this writes plot to folder
```

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

```
graphics.off() # shuts down open devices
```

Install package Hmisc and perform correlation

```
library(Hmisc)
```

```
## Loading required package: grid
## Loading required package: lattice
## Loading required package: survival
## Loading required package: splines
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:base':
##
##   format.pval, round.POSIXt, trunc.POSIXt, units
```

```
rcorr(traits$lag_time, traits$biofilm, type = "pearson")
```

```
##      x      y
## x  1.00 -0.27
## y -0.27  1.00
##
## n= 40
##
##
## P
## x      y
## x      0.0983
## y 0.0983
```

The relationship is only marginally significant ( $r = -0.27$ ,  $P = 0.098$ ). However, the relationship doesn't look linear. Use a Spearman rank correlation (non-parametric):  $r = -0.38$ ,  $P = 0.0149$ .

```
rcorr(traits$lag_time,traits$biofilm, type = "spearman")
```

```
##          x          y
## x   1.00 -0.38
## y -0.38  1.00
##
## n= 40
##
##
## P
##      x          y
## x          0.0149
## y 0.0149
```

## TESTING GROWTH RATES OF ‘OE’ AND ‘NM’ UNDER LAB CONDITIONS

Read OD600 data from cultures to calculate Malthusian growth rates

```
malth <- read.csv("data/Biofilm_Malthusian.csv",sep = ",", header=TRUE)
```

Define Time Points

```
t0 <- malth[which(malth$Time_h==0),]
t38 <- malth[which(malth$Time_h==38),]
data <- cbind(t0,t38)

data_growth <- data[ -c(1,6,8,9) ]
colnames(data_growth)[1] <- "t0"
colnames(data_growth)[4] <- "OD_t0"
colnames(data_growth)[5] <- "t38"
colnames(data_growth)[6] <- "OD_t38"
```

Calculate growth rates and conduct t-test

```
rate <- (log(data_growth$OD_t38) - log(data_growth$OD_t0)) / (data_growth$t38-data_growth$t0)
growth <- cbind(data_growth,rate)
malth.t.test <- t.test(growth$rate ~ growth$Strain)
```

No effect of strain on growth rate under laboratory maintenance conditions. Welch Two-Sample t-test:  $t = -1.118$ ,  $df = 2.217$ ,  $p\text{-value} = 0.37$

## FIGURE 1: MULTI-PANEL BIOFILM

Panel A: Comparison of ‘NM’ and ‘OE’ with O’Toole assay

Load data, calculate group means and SEM, run t-test

```

otoole <- read.csv("data/Biofilm_Otoole.csv", head = TRUE, sep = ",")
sem <- function(x) sqrt(var(x)/length(x))
strain <- factor(otoole$Strain, levels = c('non-mucoid', 'mucoid'))
otoole.means <- tapply(otoole$Biofilm, strain, mean)
otoole.sem <- tapply(otoole$Biofilm, strain, sem)
otoole.t.test <- t.test(otoole$Biofilm ~ otoole$Strain)

```

Plot biofilm data

```

# Set-up for printing
png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure1.png",
     width = 800, height = 1200, res = 96*2)

# Set plotting parameters
par(mar = c(1, 8, 4, 8))
layout <- layout(rbind(1, 2, 3), height = c(3, 0.8, 3))
#layout.show(layout)

# Make initial plot
otoole.plot <- plot(otoole.means, log = "y", ylim = c(0.05, 2.4), xlim = c(0.5, 2.5),
  pch = 22, bg = c("white", "grey"), lwd = 2, cex = 3, yaxt = "n", xaxt = "n",
  cex.lab = 2, cex.axis = 1.5, las = 1, ylab = "", xlab = "")

  box(lwd = 2)

# Add y-label axis for NM vs. OE plot
mtext(expression('Biofilm Production (a'[550]*')'), side = 2, outer = TRUE, cex = 1,
  line = -4, adj = 0.87)

# Add p-value for t-test
mtext(expression(~italic("P")~"< 0.001"), line = -1.75, cex = 1, at = 0.9)
text(2.4, 2, labels = "A", cex = 2)

# Major Axes
axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
  labels = c(0.1, 1), at = c(0.1, 1))

axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,
  at=c(0.1, 1), labels = F)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1,
  labels = c("NM", "OE"), at = c(1, 2))

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
  at = c(1, 2), labels = F)

# Minor Axes
axis(side = 2, tck = -0.015, labels = F, lwd.ticks = 2,
  at = c(0.05, 0.1, 0.25, 0.5, 2))

axis(side = 4, tck = -0.015, labels = F, lwd.ticks = 2,
  at = c(0.05, 0.1, 0.25, 0.5, 2))

```

```

# Load colony images

NM.img <- readPNG("./data/NM.png")
grid.raster(NM.img, x = 0.38, y = 0.48, height = 0.1)

OE.img <- readPNG("./data/OE.png")
grid.raster(OE.img, x = 0.61, y = 0.48, height = 0.1)

# Load biofilm density curve

kern.data <- read.csv(file = "./data/Biofilm_Spp.csv", head=TRUE, sep=",")
kern <- density(kern.data$Biofilm)

plot.new()
par(mar = c(4, 8, 2, 8))
plot(kern, main = NA, xaxt = "n", yaxt = "n", cex.lab = 1.5, ylab = "",
      xlab = "", xlim = c(-0.75, 3.8), ylim = c(0,1), lwd = 2)

mtext('Density', side = 2, outer = TRUE, cex = 1,
      line = -4, adj = 0.25)

mtext(expression('Biofilm Production (a'[550]*')'), side = 1, outer = TRUE,
      cex = 1, line = -1, adj = 0.5)

axis(side = 1, lwd.ticks = 2, cex.axis = 1.5, las = 1,
      at = c(0, 1, 2, 3), labels = T)

axis(side = 3, lwd.ticks = 2, cex.axis = 1.5, las = 1,
      at = c(0, 1, 2, 3), labels = F)

axis(side = 2, lwd.ticks = 2, cex.axis = 1.5, las = 1,
      at = c(0, 0.5, 1.0), labels = T)

axis(side = 4, lwd.ticks = 2, cex.axis = 1.5, las = 1,
      at = c(0, 0.5, 1.0), labels = F)

box(lwd=2)

arrows(0.2, 0.6, 0.2, 0.8, length = 0.05, col = "Black")
text(0.2, 0.55, labels = " NM", cex = 0.75)
arrows(3, 0.32, 3, 0.12, length = 0.05, col = "Black")
text(3, 0.36, labels = " OE", cex = 0.75)
text(3.6, 0.95, labels = "B", cex = 2)

dev.off() # this writes plot to folder

## pdf
## 2

graphics.off() # shuts down open devices

```

## FIGURE 2: SURVIVORSHIP

```
require(survival)
require(extrafont)
```

```
## Loading required package: extrafont
## Registering fonts with R
```

```
surv <- read.table("./data/Biofilm_Survivorship.txt",header=TRUE,sep="\t")
biofsurv <- Surv(surv$Days,surv$Event)
biofsurv.fit <- survfit(biofsurv ~ surv$Group, conf.int = TRUE, type = "kaplan-meier")
biofsurv.diff <- survdiff(Surv(surv$Days,surv$Event) ~ surv$Group, data = surv)
```

Plotting

```
png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure2.png",
     width = 1200, height = 1200, res = 96*2)

par(mar = c(5, 7, 5, 7))
plot(biofsurv.fit, conf.int = TRUE, mark.time = FALSE,
     xlim = c(0,50), ylim = c(0,1),
     lty = c(1,3,3,1,3,3),
     col = c("black", "grey", "grey", "black", "grey", "grey"),
     xlab = "Time (d)",
     ylab = "", cex.lab = 1.5, cex.axis = 1.2, las = 1, lwd = 2,
     yaxt = "n", xaxt = "n")
box(lwd=2)

mtext("Survivorship", side = 2, outer = TRUE, cex = 1.5, line = -3, adj = 0.5)

axis(side = 2, labels = T, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, 0.25, 0.5, 0.75, 1.0))
axis(side = 4, labels = F, lwd.ticks = 2,
     at = c(0, 0.25, 0.5, 0.75, 1.0))
axis(side = 1, labels = T, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, 10, 20, 30, 40, 50))
axis(side = 3, labels = F, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, 10, 20, 30, 40, 50))

text(44, 0.65, "OE", cex = 1.25)
text(25, 0.41, "NM", cex = 1.25)

dev.off() # this writes plot to folder

## pdf
## 2

graphics.off() # shuts down open devices
```

### FIGURE 3: GROWTH RATES VS. WATER POTENTIAL

Load data, perform multiple regressions, and calculate Psi stars

```
# Load data
growth <- read.table("./data/Biofilm_Growth.txt",header=TRUE,sep="\t")

# Specify variables
MPa <- growth[,1]
Growth <- growth[,2]
Strain <- growth[,3]

# Create Dummy Variables by Strain
D1 <- (Strain == "NM")*1
fit <- lm(Growth ~ MPa + D1 + MPa*D1)
summary(fit)

##
## Call:
## lm(formula = Growth ~ MPa + D1 + MPa * D1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.746 -0.310  0.136  0.542  1.267
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0906     0.2229   9.38 1.2e-13 ***
## MPa           2.0786     0.3370   6.17 5.2e-08 ***
## D1            -1.1869     0.3153  -3.76 0.00036 ***
## MPa:D1         0.0332     0.4766   0.07 0.94471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.756 on 64 degrees of freedom
## Multiple R-squared:  0.653, Adjusted R-squared:  0.637
## F-statistic: 40.2 on 3 and 64 DF, p-value: 1.02e-14

# Multiple regression parameters
OE.int <- fit$coefficients[1]
OE.slp <- fit$coefficients[2]
NM.int <- OE.int + fit$coefficients[3]
NM.slp <- OE.slp + fit$coefficients[4]

# Strain-specific values
NM <- growth[which(growth$Strain == "NM"),]
OE <- growth[which(growth$Strain == "OE"),]

# Plotting
png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure3.png",
     width = 1200, height = 1200, res = 96*2)
par(mar = c(7, 7, 5, 7))
plot(jitter(NM$MPa, factor = 10), NM$Growth, xlim = c(max(growth$MPa + 0.1),
```

```

min(growth$MPa - 0.2)), ylim = c(-3.5, 3),
pch = 22, bg = "white", col = "black", cex = 2,
ylab = "", xlab = "", cex.lab = 1.5, cex.axis = 1.2,
las = 1, lwd = 2, yaxt = "n", xaxt = "n")
box(lwd=2)

# Add points
points(jitter(OE$MPa, factor = 10), OE$Growth, pch = 22, bg = "grey",
       col = "black", cex = 2, lwd = 2)

# Add axis labels
mtext(expression('Growth Rate (d-1*)'), side = 2, outer = TRUE,
       cex = 1.5, line = -3, adj = 0.5)
mtext("Water Potential (-MPa)", side = 1, outer = TRUE, cex = 1.5,
       line = -3, adj = 0.5)

# Add ticks and tick labels
axis(side = 2, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c("-3.0", "-1.5", "0.0", "1.5", "3.0"), at = c(-3, -1.5, 0.0, 1.5, 3.0))

axis(side = 4, labels = F, lwd.ticks = 2,
     at = c(-3, -1.5, 0.0, 1.5, 3.0))

axis(side = 1, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c("0.0", "-0.5", "-1.0", "-1.5"), at = c(0, -0.5, -1, -1.5))

axis(side = 3, labels = F, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, -0.5, -1, -1.5))

# Add Psi star lines
NM.star <- -NM.int/NM.slp
segments(x0 = NM.star, y0 = -3.3, x1 = NM.star, y1 = 0, col = "black",
        lwd = 2.5, lty = 6, xpd = T)

OE.star <- -OE.int/OE.slp
segments(x0 = OE.star, y0 = -3.3, x1 = OE.star, y1 = 0, col = "black",
        lwd = 2.5, lty = 4, xpd = T)

# Add zero-growth line
abline(h = 0, col = "black", lty = 3, lwd = 2)

# Add multiple regression lines
clip(0.05,-1.6,-3.5,3)
abline(a = NM.int, b = NM.slp, col = "black", lwd = 2.5, lty = 6)

clip(0.05,-1.6,-3.5,3)
abline(a = OE.int, b = OE.slp, col = "black", lwd = 2.5, lty = 4)

# Add legend
legend(-1.2, 3.24, c("OE", "NM"), pch = 22, pt.bg = c("grey", "white"), pt.cex = 2,
      pt.lwd = 2, bty = 'n', y.intersp = 1, lty = c(6,4), lwd = 2.5)

# Add Psi star symbols to plot

```



```

mtext(side = 1, line = -1, at = NM.star, bquote(psi~'*'[NM]), cex = 1.2)
mtext(side = 1, line = -1, at = OE.star, bquote(psi~'*'[OE]), cex = 1.2)

# Add R2
text(-0.075, 2.9, labels = expression(R^2 == 0.65), cex = 1)

dev.off() # this writes plot to folder

```

```

## pdf
## 2

```

```

graphics.off() # shuts down open devices

```

## FIGURE 4: RESPIRATION VS. WATER POTENTIAL

Load data and run MLE analyses for multiple model comparisons

```

resp.raw <- read.csv("./data/Biofilm_Respiration.csv", header = TRUE, sep = ",")
resp.trunc <- resp.raw[,1:3] # gets rid of trailing data
resp.uneg <- resp.raw[,1]*-1 # convert MPa to positive values
resp <- data.frame(resp.uneg, resp.trunc) # add uneg to dataframe
colnames(resp)[1] <- "Wp" # "Wp" = uneg MPa (positive)
require(bbmle)

```

```

## Loading required package: bbmle
## Loading required package: stats4

```

```

#starting values for niche model
A = 35 # Maximum respiratoin
X = 0 # Optimum MPa
B = 0.05 # Niche Breadth
T = 1 # Tau, shape kernel
Z = 7.5 # Error

```

```

fit1 <- mle2(R ~ dnorm(mean = a, sd = z), start = list(a = A, z = Z), data = resp)

```

```

## Warning: NaNs produced

```

```

# null

fit2 <- mle2(R ~ dnorm(mean = a * exp(-((abs(W + x))/b)^t), sd = z),
  method = "L-BFGS-B", lower = c(x = 0.0000, b = 0.001),
  start = list(a = A, x = X, b = B, t = T, z = Z), data = resp)

```

```

## Warning: longer object length is not a multiple of shorter object length
## Warning: convergence failure: code=1 (NEW_X)

```

```

# common parameter set for both strains

fit3 <- mle2(R ~ dnorm(mean = a * exp(-((abs(W + x))/b)^t), sd = z),
  method = "L-BFGS-B", lower = c(x = 0.0000, b = 0.001),
  start = list(a = A, x = X, b = B, t = T, z = Z), data = resp,
  parameters = c(a ~ Strain))

```

```

## Warning: some parameters are on the boundary: variance-covariance
## calculations based on Hessian may be unreliable

```

```

# max differs between strains

fit4 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(x ~ Strain))
# opt water differs between strain

fit5 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(b ~ Strain))
# breadth differs between strains

fit6 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(t ~ Strain))

```

```

## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may
## Warning: convergence failure: code=1 (NEW_X)

```

```

# tau differs between strains

fit7 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, x ~ Strain))

```

```

## Warning: longer object length is not a multiple of shorter object length
## Warning: convergence failure: code=1 (NEW_X)

```

```

# max and opt differ between strains

fit8 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, b ~ Strain))

```

```

## Warning: longer object length is not a multiple of shorter object length
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may

```

```

# max and breadth differ between strains

fit9 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, t ~ Strain))

```

```
## Warning: longer object length is not a multiple of shorter object length
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may be unreliable
```

```
# max and tau differ between strains
```

```
fit10 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(x ~ Strain, b ~ Strain))
```

```
## Warning: longer object length is not a multiple of shorter object length
```

```
# opt and breadth differ between strains
```

```
fit11 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(x ~ Strain, t ~ Strain))
```

```
## Warning: longer object length is not a multiple of shorter object length
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may be unreliable
```

```
# opt and tau differ between strains
```

```
fit12 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(b ~ Strain, t ~ Strain))
```

```
## Warning: longer object length is not a multiple of shorter object length
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may be unreliable
```

```
# breadth and tau differ between strains
```

```
fit13 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, x ~ Strain,
  b ~ Strain))
```

```
## Warning: some parameters are on the boundary: variance-covariance
## calculations based on Hessian may be unreliable
```

```
# max, opt, breadth differ between strains
```

```
fit14 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, x ~ Strain,
  t ~ Strain))
```

```
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may be unreliable
## Warning: convergence failure: code=1 (NEW_X)
```

```

# max, opt, tau differ between strains

fit15 <- mle2(R ~ dnorm(mean= a*exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B",
  lower = c(x = 0.0000, b = 0.001), start = list(a = A, x = X, b = B,
  t = T, z = Z), data = resp, parameters = c(a ~ Strain, b ~ Strain,
  t ~ Strain))

```

```

## Warning: some parameters are on the boundary: variance-covariance
## calculations based on Hessian may be unreliable

```

```

# max, breadth, tau differ between strains

fit16 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z),
  method = "L-BFGS-B", lower = c(x = 0.0000, b = 0.001),
  start = list(a = A, x = X, b = B, t = T,
  z = Z), data = resp, parameters = c(x ~ Strain,
  b ~ Strain, t ~ Strain))

```

```

## Warning: some parameters are on the boundary: variance-covariance
## calculations based on Hessian may be unreliable

```

```

# opt, breadth, tau differ between strains

fit17 <- mle2(R ~ dnorm(mean = a*exp(-((abs(W + x))/b)^t), sd = z),
  method = "L-BFGS-B", lower = c(x = 0.0000, b = 0.001),
  start = list(a = A, x = X, b = B, t = T, z = Z),
  data = resp, parameters = c(a ~ Strain, x ~ Strain,
  b ~ Strain, t ~ Strain))

```

```

## Warning: longer object length is not a multiple of shorter object length
## Warning: some parameters are on the boundary: variance-covariance calculations based on Hessian may be unreliable
## Warning: convergence failure: code=1 (NEW_X)

```

```

#all params differ between strains

AIC.table <- AICtab(fit1, fit2, fit3, fit4, fit5, fit6, fit7, fit8, fit9, fit10,
  fit11, fit12, fit13, fit14, fit15, fit16, fit17, delta = T, weights = T)
AIC.names <- attributes(AIC.table)$row.names
AIC.table <- as.data.frame(do.call(cbind, c(AIC.table)))
AIC.table$names <- AIC.names
write.table(AIC.table, "~/GitHub/Biofilms/figures/AIC.table.txt", sep = "\t")

# Conclusion from based on AIC is that the following models all perform equally
# well: fit10, fit 4, fit7, fit16, fit13, and fit11.
# However, all of these models contain Wopt. Therefore, we use fit4, which only
# distinguishes OE and NM based on Wopt.

```

## Calculate niche breadth

Based on Lennon et al. (2012), we estimate niche breadth (nb) using the *b* parameter from the equation above.

```

# Following calculated estimates and error for 'b', which is sigma in model
NM.sig <- fit4@coef[2]
NM.sig.e <- 0.022994 # con't figure out how to extract Std. Error from fit4
OE.sig <- fit4@coef[3] + fit4@coef[2]
OE.sig.e <- NM.sig.e + 0.039542

# Use sigma values to estimate 'nb'
x.Rmax <- 0.5 # percent of Rmax defining niche breadth

NM.nb <- NM.sig * (-log10(x.Rmax))^(1/fit4@coef[5]) # niche breadth for NM
NM.nb.e <- NM.nb/(NM.sig/NM.sig.e) # SEM niche breadth for NM

OE.nb <- OE.sig * (-log10(x.Rmax))^(1/fit4@coef[5]) # niche breadth for OE
OE.nb.e <- OE.nb/(OE.sig/OE.sig.e) # SEM niche breadth for OE

```

## Plot Data

```

OE.resp <- subset(resp[,2:3], resp[,4] == "OE", data = resp)
NM.resp <- subset(resp[,2:3], resp[,4] == "NM", data = resp)

png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure4.png",
     width = 1200, height = 1200, res = 96*2)

plot.new()
par(mar = c(7, 7, 5, 7))

plot(jitter(OE.resp[,1], factor = 10), OE.resp[,2], xlim = c(0.5, -3.5),
     ylim = c(-2.5, 40), type = "p",
     pch = 22, bg = "grey", col = "black", cex = 2, ylab = "", xlab = "",
     cex.lab = 1.5, las = 1, lwd = 2, yaxt = "n", xaxt = "n")
box(lwd=2)

points(jitter(NM.resp[,1], factor = 10), NM.resp[,2], type = "p", pch = 22,
       bg = "white", col = "black",
       cex = 2, cex.lab = 1.5, lwd = 2)

# Add ticks and tick labels
axis(side = 2, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c(0, 10, 20, 30, 40), at = c(0, 10, 20, 30, 40))

axis(side = 4, labels = F, lwd.ticks = 2,
     at = c(0, 10, 20, 30, 40))

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1, mgp = c(3, 1, 0),
     labels = c(0, -1, -2, -3, -4), at = c(0, -1, -2, -3, -4))

axis(side = 3, labels = F, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, -1, -2, -3, -4))

mtext('Water Potential (-MPa)', side = 1, outer = TRUE, cex = 1.5,
     line = -4, adj = 0.5)

```

```

mtext(expression(paste('Respiration (', mu , 'gC-CO'[2]* ' g soil'^-1* 'd'^-1*')),
        side = 2, outer = TRUE, cex = 1.5, line = -3.5, adj = 0.6)

legend(-2, 40, c("OE", "NM"), pch = 22, pt.bg = c("grey", "white"), pt.cex = 2,
        pt.lwd = 2, bty = 'n', y.intersp = 1, lty = c(4,6), lwd = 2.5, seg.len = 5)

# Add functions to plot

# First curve is for NM strain
# curve(coef(fit10)[[1]]*exp(-((abs(coef(fit10)[[2]]+x)/coef(fit10)[[4]]))^coef(fit10)[[6]]),
#       from = 0.2, to = -3.3, add = TRUE, lty = 6, lwd = 2.5) # adds NM
curve(coef(fit4)[[1]]*exp(-((abs(coef(fit4)[[2]]+x)/coef(fit4)[[4]]))^coef(fit4)[[5]]),
      from = 0.2, to = -3.3, add = TRUE, lty = 6, lwd = 2.5) # adds NM

# For second curve, need to modify to get OE strain parameters
# OEx=coef(fit10)[2]+coef(fit10)[3] # modifies optimum parameter relative to NM
# OEb=coef(fit10)[4]+coef(fit10)[5] # modified breadth parameter relative to NM
OEx=coef(fit4)[2]+coef(fit4)[3] # modifies optimum parameter relative to NM

curve(coef(fit10)[[1]]*exp(-((abs(OEx+x)/coef(fit4)[[4]]))^coef(fit4)[[5]]),
      from = 0.2, to = -3.3, add = TRUE, lty = 4, lwd = 2.5) # adds OE

dev.off()

## pdf
## 2

graphics.off()

```

```

fit2 <- mle2(R ~ dnorm(mean = a * exp(-((abs(W + x))/b)^t), sd = z), method = "L-BFGS-B", lower = c(x
= 0.0000, b = 0.001), start = list(a = A, x = X, b = B, t = T, z = Z), data = resp) # common parameter
set for both strains

```

## FIGURE 5: REWETTING COLUMN EXPERIMENT

### Read and arrange data

```

cols <- read.csv("../data/Biofilm_Columns.csv", header = TRUE, sep = ",")
head(cols)

```

```

##   Column day      OE.1      OE.2      OE.3      OE.4      NM.1
## 1   day0    0 -0.0001135 -0.0001135 -0.0001135 -0.0001135 -0.0001135
## 2   day2    2 -0.0001132 -0.0001132 -0.0001105 -0.0001141 -0.0001139
## 3   day3    3 -0.0001135 -0.0001132 -0.0001131 -0.0001138 -0.0001137
## 4   day6    6 -0.0001132 -0.0001132 -0.0001132 -0.0001138 -0.0001141
## 5   day8    8 -0.0001138 -0.0001132 -0.0001135 -0.0001137 -0.0001141
## 6  day13   13 -0.0001138 -0.0001131 -0.0001134 -0.0001144 -0.0001142
##              NM.2      NM.3      NM.4
## 1 -0.0001135 -0.0001135 -0.0001135

```

```
## 2 -0.0001141 -0.0001141 -0.0001139
## 3 -0.0001134 -0.0001138 -0.0001131
## 4 -0.0001137 -0.0001139 -0.0001134
## 5 -0.0001131 -0.0001139 -0.0001134
## 6 -0.0001135 -0.0001141 -0.0001137
```

```
sem <- function(x) sqrt(var(x)/length(x))

OE.cols <- data.frame(cols[,2], apply(cols[, 3:6], 1, mean),
                      apply(cols[, 3:6], 1, sem))
colnames(OE.cols)[1:3] <- c("day", "OE.mean", "OE.sem")

NM.cols <- data.frame(cols[,2], apply(cols[, 7:10], 1, mean),
                      apply(cols[, 7:10], 1, sem))
colnames(NM.cols)[1:3] <- c("day", "NM.mean", "NM.sem")
```

## Make main plot

```
png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure5.png",
     width = 1200, height = 800, res = 96*2)

plot.new()
par(mar = c(7, 7, 5, 7))

plot(OE.cols[,1], OE.cols[,2], xlim = c(0,100), ylim = c(-0.017, 0.001), type = "b",
     pch = 22, bg = "grey", col = "black", cex = 2, ylab = "", xlab = "",
     cex.lab = 1.5, las = 1, lwd = 2, yaxt = "n", xaxt = "n",
     panel.first={
       arrows(x0 = OE.cols[,1], y0 = OE.cols[,2], y1 = OE.cols[, 2] - OE.cols[, 3],
             angle = 90, length = 0.05, lwd = 2)
       arrows(x0 = OE.cols[,1], y0 = OE.cols[,2], y1 = OE.cols[, 2] + OE.cols[, 3],
             angle = 90,
             length = 0.05, lwd = 2)
     })
```

```
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
```

```
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
```

```
box(lwd=2)

# Construct NM plot
points(NM.cols[,1], NM.cols[,2], type = "b", pch = 22, bg = "white", col = "black",
       cex = 2, cex.lab = 1.5, lwd = 2,
       panel.first={
         arrows(x0 = NM.cols[,1], y0 = NM.cols[, 2], y1 = NM.cols[, 2] - NM.cols[, 3],
               angle = 90, length = 0.05, lwd = 2)
         arrows(x0 = NM.cols[,1], y0 = NM.cols[, 2], y1 = NM.cols[, 2] + NM.cols[, 3],
               angle = 90, length = 0.05, lwd = 2)
         abline(v = c(23, 46, 67), col = "black", lty = 3, lwd = 2)
       })
```

```
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
## Warning: zero-length arrow is of indeterminate angle and so skipped
```

```
# Add ticks and tick labels
axis(side = 2, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c("0.000", "-0.005", "-0.010", "-0.015"),
     at = c(0, -0.005, -0.010, -0.015))

axis(side = 4, labels = F, lwd.ticks = 2,
     at = c(0, -0.005, -0.010, -0.015))

axis(side = 1, lwd.ticks = 2, cex.axis = 1.25, las = 1, mgp = c(3, 2, 0),
     labels = c(0, 25, 50, 75, 100), at = c(0, 25, 50, 75, 100))

axis(side = 3, labels = F, lwd.ticks = 2, las = 1, cex.axis = 1.25,
```



```

    at = c(0, 25, 50, 75, 100))

mtext('Water Potential (-MPa)', side = 2, outer = TRUE, cex = 1.5,
      line = -2, adj = 0.6)

mtext('Time (d)', side = 1, outer = TRUE, cex = 1.5,
      line = -2.5, adj = 0.5)

# Add in-figure caption
text(97, -0.005, labels = "OE", cex = 1.2)
text(97, -0.012, labels = "NM", cex = 1.2)

dev.off()

## pdf
## 2

graphics.off()

```

Test whether Pseudomonas densities are different at end of experiment

```

# Load data
cols.cfu <- read.csv("./data/Biofilm_Columns_CFU.csv", head = TRUE, sep = ",")

# Covert CFU of extraction volume to g of soil
vol <- 10 # extraction volume (ml)
soil <- 2.5 # soil extracted (g)
CFUadj <- (cols.cfu$CFU*vol)/soil
CFU <- cbind(cols.cfu, CFUadj)

cols.ttest <- t.test(CFU$CFUadj ~ CFU$Strain)
cols.ttest <- t.test(CFU$CFUadj ~ CFU$Strain, var.equal = TRUE)
stderr <- function(x) sqrt(var(x)/length(x))

OE.cols.CFU <- CFU[which(CFU$Strain=="OE"),]
SEM.OE.cols.CFU <- stderr(OE.cols.CFU$CFUadj)

NM.cols.CFU <- CFU[which(CFU$Strain=="NM"),]
SEM.NM.cols.CFU <- stderr(NM.cols.CFU$CFUadj)

```

## FIGURE 6: SPECIES INTERACTIONS VIA TESTER STRAIN

Read and arrange data

```
# Load data
comp <- read.table("./data/Biofilm_Competition.txt", sep="\t", header=TRUE)
head(comp)
```

```
##           MPa      t0      tf Hours  Strain Culture
## 1 -0.09949 1940000 7200000    96 KBS0701    mono
## 2 -0.09949 1440000 8450000    96 KBS0701    mono
## 3 -0.09949 1240000 5820000    96 KBS0701    mono
## 4 -0.48553 805000 5200000    96 KBS0701    mono
## 5 -0.48553 455000 3500000    96 KBS0701    mono
## 6 -0.48553 1660000 2450000    96 KBS0701    mono
```

```
# Calculate growth rates
rate <- (log(comp$tf)-log(comp$t0))/(comp$Hours)
growth <- cbind(comp,rate)
```

## Conduct dummy variables multiple regression (method 1)

```
# subsets KBS0701 growth rate data:
KBS0701 <- growth[which(growth$Strain == "KBS0701"),]
# dummy var for intercept (D1) and slope (MPa*D1) with 0406:
D1 <- (KBS0701$Culture == "cocult_OE")*1
# dummy var for intercept (D2) and slope (MPa*D2) on growth with 0407:
D2 <- (KBS0701$Culture!="mono")*1
# merges KBS0701 data with dummy variables:
KBS0701.2 <- cbind(KBS0701, D1, D2)
# dummy variables multiple regression:
KBS0701.fit <- lm(rate ~ MPa + D1 + MPa*D1 + D2 + MPa*D2, data = KBS0701.2)
summary(KBS0701.fit)
```

```
##
## Call:
## lm(formula = rate ~ MPa + D1 + MPa * D1 + D2 + MPa * D2, data = KBS0701.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.02836 -0.00894 -0.00181  0.01003  0.04314
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.03542    0.00846   4.19 0.00025 ***
## MPa           0.05604    0.00927   6.04 1.6e-06 ***
## D1            0.00019    0.01797   0.01 0.99166
## D2           -0.01762    0.01406  -1.25 0.22052
## MPa:D1       -0.00136    0.01678  -0.08 0.93575
## MPa:D2       -0.00888    0.01420  -0.63 0.53678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0162 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.743, Adjusted R-squared:  0.697
## F-statistic: 16.2 on 5 and 28 DF,  p-value: 1.66e-07
```

```
# Multiple regression parameters corresponding Psi stars:
tester.int <- KBS0701.fit$coefficients[1]
tester.slp <- KBS0701.fit$coefficients[2]
tester.star <- tester.int/tester.slp*-1

tester.NM.int <- tester.int + KBS0701.fit$coefficients[3]
tester.NM.slp <- tester.slp + KBS0701.fit$coefficients[5]
tester.NM.star <- tester.NM.int/tester.NM.slp*-1

tester.OE.int <- tester.int + KBS0701.fit$coefficients[4]
tester.OE.slp <- tester.slp + KBS0701.fit$coefficients[6]
tester.OE.star <- tester.OE.int/tester.OE.slp*-1

# Dummy variables regresssion: R2 = 0.74, F(5,28) = 16.2, P < 0.0001
# Intercept and Mpa are both significant (0.0003 and <0.0001)
# But no effect of dummy variables on intercepts or slopes (P > 0.22)
```

Simple linear regression for “global” model (since no effect of strain in multiple regression)

```
global.fit <- lm(KBS0701$rate ~ KBS0701$MPa)
global.int <- global.fit$coefficients[1]
global.slp <- global.fit$coefficients[2]
global.psi <- global.int/global.slp*-1
```

## Plot of growth rates

```
# Strain-specific values
solo <- KBS0701[which(KBS0701$Culture == "mono"),]
with.NM <- KBS0701[which(KBS0701$Culture == "cocult_NM"),]
with.OE <- KBS0701[which(KBS0701$Culture == "cocult_OE"),]

# Main plot
png(filename="/Users/lennonj/GitHub/Biofilms/figures/Figure6.png",
     width = 1200, height = 1200, res = 96*2)
par(mar = c(7, 7, 5, 7))
plot(jitter(solo$MPa, factor = 3), solo$rate, xlim = c(0.25, -1.85),
     ylim = c(-0.1, 0.065),
     pch = 22, bg = "black", col = "black", cex = 2,
     ylab = "", xlab = "", cex.lab = 1.5, cex.axis = 1.2,
     las = 1, lwd = 2, yaxt = "n", xaxt = "n")
box(lwd=2)

# Add points
```

```

points(jitter(with.NM$MPa, factor = 3), with.NM$rate, pch = 22,
       bg = "white", col = "black", cex = 2, lwd = 2)
points(jitter(with.OE$MPa, factor = 3), with.OE$rate, pch = 22, bg = "grey",
       col = "black", cex = 2, lwd = 2)

# Add axis labels
mtext(expression('Growth Rate (d-1*)'), side = 2, outer = TRUE,
       cex = 1.5, line = -3, adj = 0.5)
mtext("Water Potential (-MPa)", side = 1, outer = TRUE, cex = 1.5,
       line = -3, adj = 0.5)

# Add ticks and tick labels
axis(side = 2, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c("-0.10", "-0.05", "0.00", "0.05"), at = c(-0.1, -0.05, 0.0, 0.05))

axis(side = 4, labels = F, lwd.ticks = 2,
     at = c(-0.1, -0.05, 0.0, 0.05))

axis(side = 1, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     labels = c("0.0", "-0.5", "-1.0", "-1.5"), at = c(0, -0.5, -1, -1.5))

axis(side = 3, labels = F, lwd.ticks = 2, las = 1, cex.axis = 1.25,
     at = c(0, -0.5, -1, -1.5))

# Add legend
legend(-0.85, 0.07, c("tester", "tester + NM", "tester + OE"), pch = 22,
     pt.bg = c("black", "white", "grey"), pt.cex = 2, pt.lwd = 2,
     bty = 'n', y.intersp = 1, lty = c(6,4), lwd = 2.5)

# Add zero-growth line
abline(h = 0, col = "black", lty = 3, lwd = 2)

# # Add multiple regression lines (not printed; instead using "global"
# # parameters because strain parameters are not significant)
# clip(0.05,-1.6,-0.1,0.05)
# abline(a = tester.int, b = tester.slp, col = "black", lwd = 2.5, lty = 6)
#
# clip(0.05,-1.4,-3.5,3)
# abline(a = tester.NM.int, b = tester.NM.slp, col = "red", lwd = 2.5, lty = 4)
#
# clip(0.05,-1.3,-3.5,3)
# abline(a = tester.OE.int, b = tester.OE.slp, col = "blue", lwd = 2.5, lty = 4)

clip(0.1, -1.75, -0.1, 0.05)
abline(a = global.int, b = global.slp, col = "black", lwd = 2.5, lty = 6)

# # Add Psi star lines
segments(x0 = global.psi, y0 = -0.095, x1 = global.psi, y1 = 0, col = "black",
        lwd = 2.5, lty = 6, xpd = T)

# Add Psi star symbols to plot
mtext(side = 1, line = -1, at = global.psi, bquote(psi~'*'), cex = 1.2)

```

```
# Add R2
text(0, 0.06, labels = expression(R2 == 0.74), cex = 1)

dev.off()
```

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

```
graphics.off()
```