

Yampolsky et al. (2015) Adaptive phenotypic plasticity and local adaptation for temperature tolerance in freshwater zooplankton

Tad Dallas

Contents

Introduction	2
Data preprocessing	2
Figure 1	3
Table 2	4
Figure 2	5
Table 3	6
Figure 3	8
Table 4	11

Introduction

```
# find all csv files in data directory, and read files in as list elements
data <- lapply(paste('../data/', dir('../data', '\\.csv'), sep=''), read.csv,
              stringsAsFactors=FALSE)
# make colnames consistent
colnames(data[[2]])[1] <- 'Clone'

names(data) <- c('HbPublic', 'SuppTable1', 'DmagnaPublic')
```

Data preprocessing

Clone names are not conserved across datasets. Also, Clone and CloneID are two columns that pertain to the same information. Also, clone names don't sync up, which may be the reason for the lack of reproducibility for Tables 2 and 4.

```
data[[3]][which(data[[3]][, 'Clone'] == 'Xinb3'), 'Clone'] <- 'Xinb-3'
data[[3]][which(data[[3]][, 'Clone'] == 'Iinb1'), 'Clone'] <- 'Iinb-1'

aht <- c(23, 25, 20, NA, NA, 21, 21, 27, 23, 21, 24, 31, 30, 30,
        24, 20, 17, 17, 23, 25, 21, 25, 20)

names(aht) <- c("BE-OM-2", "CH-H-149", "EL-75-69", "ELNA-1", "F1", "FAT-1-6",
               "FHS-2-11-8", "FRC-1-1", "FSP-1-16-2", "FUT-1-2-1", "Iinb-1",
               "IR-1-7", "ISR-1-8", "Israel-8", "Mu-11", "N-47-20", "RBo1-1-1",
               "RKOR-1-1", "RMZ-009", "RYAK", "SE-G1-9", "WR-G1-9", "Xinb-3")
```

Figure 1

The results are qualitatively similar, but there were some clones that I could not match the names of, as naming of clones differed between data tables (see `data[[2]]` and `data[[3]]`).

```
getFig1 <- function(data, aht, plotIt=TRUE){
  clone <- unique(data[[3]]$Clone)
  retHot <- matrix(0, ncol=2, nrow=length(clone))
  retCold <- matrix(0, ncol=2, nrow=length(clone))
  # loops through clones and determines mean and SE for
  # each clone, and each acclimation temperature treatment

  for(i in 1:length(clone)){
    retHot[i,1] <- mean(data[[3]][which(data[[3]]$Clone == clone[i]
                                         & data[[3]]$AccT == 28), 'logTimm'])

    retHot[i,2] <- sd(data[[3]][which(data[[3]]$Clone == clone[i] &
                                     data[[3]]$AccT == 28), 'logTimm']) /
      sqrt(length(data[[3]][which(data[[3]]$Clone == clone[i] &
                                   data[[3]]$AccT == 28), 'logTimm'])))

    retCold[i,1] <- mean(data[[3]][which(data[[3]]$Clone == clone[i] &
                                          data[[3]]$AccT == 20), 'logTimm'])

    retCold[i,2] <- sd(data[[3]][which(data[[3]]$Clone == clone[i] &
                                       data[[3]]$AccT == 20), 'logTimm']) /
      sqrt(length(data[[3]][which(data[[3]]$Clone == clone[i] &
                                   data[[3]]$AccT == 20), 'logTimm'])))
  }

  if(plotIt == TRUE){
    plot(aht, retHot[,1], pch=21, las=1, ylim=c(3.5, 5.5), tck=0.01, bg = 'white',
         ylab= expression(paste('log(T' [imm],')')),
         xlab=expression(paste('warmest month average high, T (', ~degree , 'C)'))))

    segments(x0 = aht, y0 = retHot[,1]+retHot[,2], y1 = retHot[,1] - retHot[,2])
    abline(lm(retHot[,1] ~ aht), lwd=2, lty=2 )
    points(aht, retHot[,1], pch = 21, bg='white')

    points(aht, retCold[,1], pch = 16)
    segments(x0 = aht, y0 = retCold[,1]+retCold[,2], y1 = retCold[,1] - retCold[,2])
    abline(lm(retCold[,1] ~ aht), lwd=2)}

  if(plotIt == FALSE){
    ret <- rbind(retHot, retCold)
    acct <- c(rep(28, nrow(retHot)), rep(20, nrow(retCold)))
    return(aov(glm(ret[,1] ~ rep(aht, 2) * acct)))}
}

getFig1(data, aht, plotIt=TRUE)
```

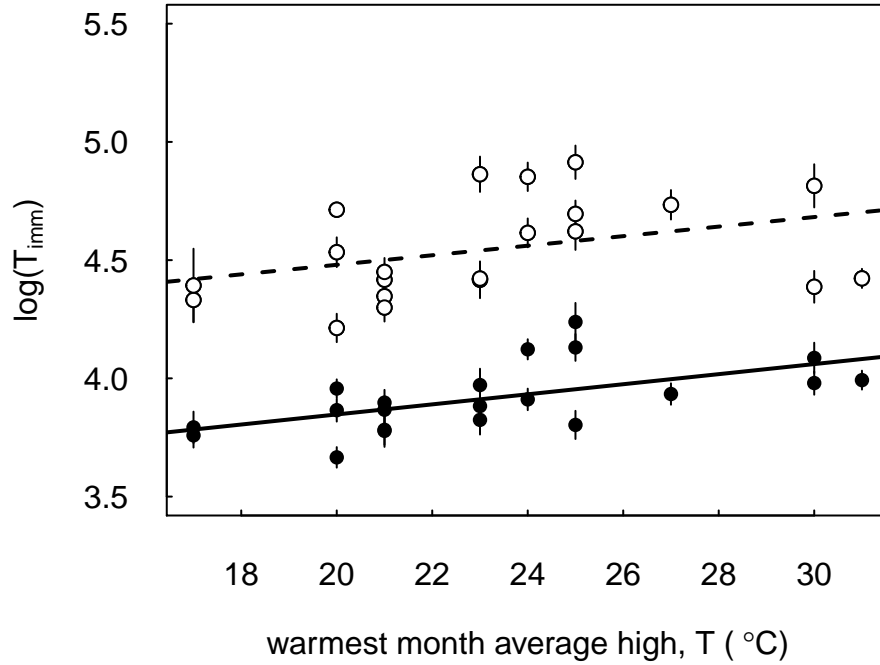


Figure 1: Time until immobilization as a function of the warmest month average high.

Table 2

This is all wrong, because of the data issues described above. Whenever I get the clone naming issues figured out, the below table should be similar to Table 1.

```
library(xtable)
glmTimm <- getFig1(data, aht, plotIt=FALSE)
glmTimmTable <- matrix(unlist(summary(glmTimm)), ncol=5)
rownames(glmTimmTable) <- c("AHT_warmest", "AccT", "Acct * AHT_warmest", "Residuals")
colnames(glmTimmTable) <- c('df', 'SS', 'MS', 'F', 'p-value')
print(xtable(glmTimmTable), floating.placement='!' )
```

% latex table generated in R 3.1.1 by xtable 1.7-4 package % Thu Aug 27 22:25:35 2015

	df	SS	MS	F	p-value
AHT_warmest	1.00	0.26	0.26	9.84	0.00
AccT	1.00	4.16	4.16	155.31	0.00
Acct * AHT_warmest	1.00	0.00	0.00	0.01	0.93
Residuals	38.00	1.02	0.03		

Figure 2

I took a few liberties and added some pretty colors instead of the cross hatch black and white from the original paper.

```
getFig2 <- function(data, color = rainbow(4), ...){
  require(dplyr)
  short <- data[[3]][which(data[[3]][, 'Experiment'] == 'Short-term feeding'),]
  long <- data[[3]][which(data[[3]][, 'Experiment'] == 'Long-term feeding'),]

  shBarData <- short %>%
    group_by(Treatment, AccT) %>%
    summarize(mean(logTimm), sd(logTimm)/sqrt(length(logTimm)))

  lgBarData <- long %>%
    group_by(Treatment, AccT) %>%
    summarize(mean(logTimm), sd(logTimm)/sqrt(length(logTimm)))

  layout(matrix(c(1,2),ncol=2))
  shBar <- barplot(shBarData[[3]][c(3,1,4,2)], space = c(0.2, 0, 1, 0),
    las=1, tck=0.01, col=color, ylab=expression(paste('log(T'[imm],')')),
    xlab=expression(paste('acclimitization temperature (', ~degree , 'C)')),
    ...)
  legend('topright', expression(italic('a')), bty='n', cex=1.25)
  axis(1, labels=c('20', '28'), at = c(1.2, 4.6), tck=0.01)

  segments(x0 = shBar,
    y0 = shBarData[[3]][c(3,1,4,2)] + shBarData[[4]][c(3,1,4,2)],
    y1 = shBarData[[3]][c(3,1,4,2)] - shBarData[[4]][c(3,1,4,2)])

  lgBar <- barplot(lgBarData[[3]][c(3,1,4,2)], space = c(0.2, 0, 1, 0),
    las=1, tck=0.01, col=color, ylab='',
    xlab=expression(paste('acclimitization temperature (', ~degree , 'C)')),
    ...)
  legend('topright', expression(italic('b')), bty='n', cex=1.25)
  axis(1, labels=c('20', '28'), at = c(1.2, 4.6), tck=0.01)

  segments(x0 = lgBar,
    y0 = lgBarData[[3]][c(3,1,4,2)] + lgBarData[[4]][c(3,1,4,2)],
    y1 = lgBarData[[3]][c(3,1,4,2)] - lgBarData[[4]][c(3,1,4,2)])
}

prettyColors <- c('#d9d9f9', '#a5cd26', '#ed9fdb', '#cd26a5')
getFig2(data, color = prettyColors, xpd=FALSE, ylim=c(3,5.5))
```

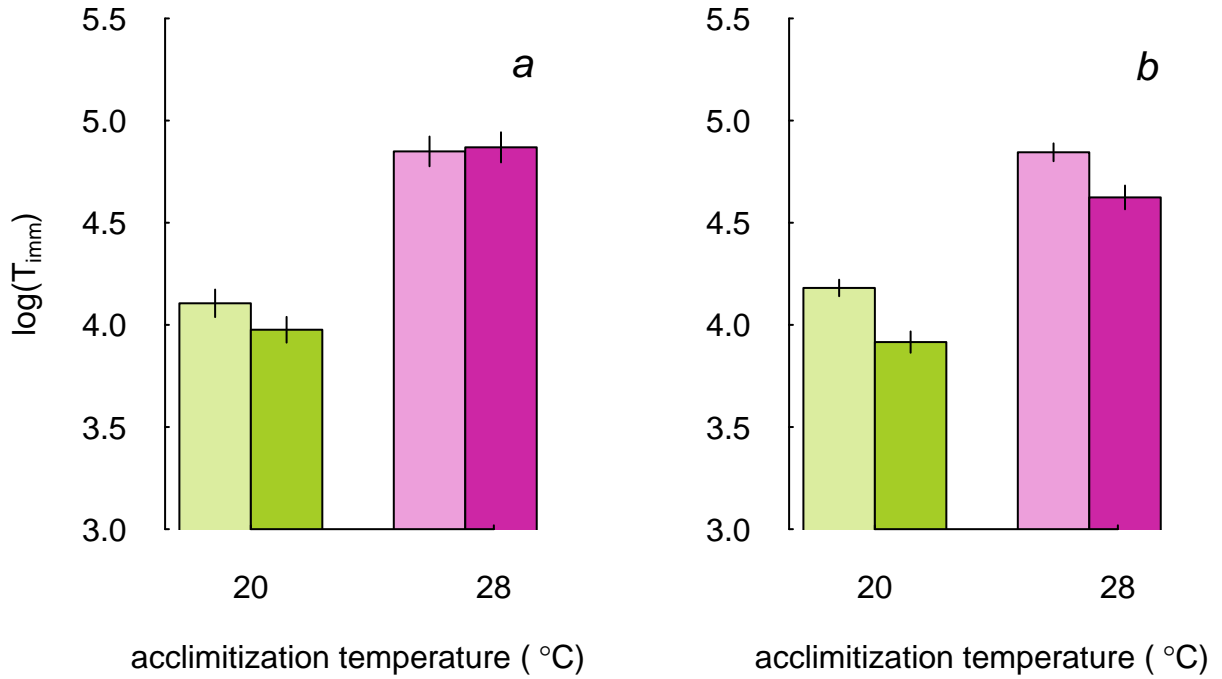


Figure 2: Time until immobilization as a function of acclimatization temperature (20 or 28 °C) under both high (darker bars) and limited (lighter bars) food resource levels for both short-term (8h; panel *a*) and long-term (1 generation; panel *b*) nutritional manipulation treatments.

Table 3

```
getTable3 <- function(data){
  require(xtable)
  datS <- data[[3]][which(data[[3]]$Experiment == 'Short-term feeding'),]
  datL <- data[[3]][which(data[[3]]$Experiment == 'Long-term feeding'),]

  glmS <- aov(glm(datS$logTimm ~ datS$AccT * datS$Treatment))
  glmS2 <- matrix(unlist(summary(glmS)), ncol=5)
  rownames(glmS2) <- c("AccT", "food", "AccT * food", "Residuals")
  colnames(glmS2) <- c('df', 'SS', 'MS', 'F', 'p-value')

  glmL <- aov(glm(datL$logTimm ~ datL$AccT * datL$Treatment))
  glmL2 <- matrix(unlist(summary(glmL)), ncol=5)
  rownames(glmL2) <- c("AccT ", "food ", "AccT * food ", "Residuals ")
  colnames(glmL2) <- c('df', 'SS', 'MS', 'F', 'p-value')

  glmTable <- rbind(rep(NA, 5), glmS2, rep(NA, 5), glmL2)

  rownames(glmTable)[c(1,6)] <- c("short-term food manipulation treatment",
                                "long-term food manipulation treatment")

  print(xtable(glmTable), floating.placement='!',
        comment = FALSE, digits = 2)
}
```

```
getTable3(data)
```

	df	SS	MS	F	p-value
short-term food manipulation treatment					
AccT	1.00	8.03	8.03	138.92	0.00
food	1.00	0.04	0.04	0.62	0.43
Acct * food	1.00	0.07	0.07	1.16	0.29
Residuals	44.00	2.54	0.06		
long-term food manipulation treatment					
AccT	1.00	5.65	5.65	195.40	0.00
food	1.00	0.71	0.71	24.50	0.00
Acct * food	1.00	0.01	0.01	0.20	0.66
Residuals	44.00	1.27	0.03		

Figure 3

```
getFig3 <- function(data, aht, plotIt=TRUE, ...){
  clone <- unique(data[[3]]$Clone)
  retHot <- matrix(0, ncol=2, nrow=length(clone))
  h576Hot <- matrix(0, ncol=2, nrow=length(clone))
  retCold <- matrix(0, ncol=2, nrow=length(clone))
  h576Cold <- matrix(0, ncol=2, nrow=length(clone))

  # loops through clones and determines mean and SE for
  # each clone, and each acclimation temperature treatment
  # a dplyr solution would be easy (I do it somewhere in here)

  for(i in 1:length(clone)){
    retHot[i,1] <- mean(data[[3]][which(data[[3]]$Clone == clone[i]
                                         & data[[3]]$AccT == 28), 'logTimm'])

    retHot[i,2] <- sd(data[[3]][which(data[[3]]$Clone == clone[i] &
                                      data[[3]]$AccT == 28), 'logTimm']) /
      sqrt(length(data[[3]][which(data[[3]]$Clone == clone[i] &
                                   data[[3]]$AccT == 28), 'logTimm'])))

    retCold[i,1] <- mean(data[[3]][which(data[[3]]$Clone == clone[i] &
                                          data[[3]]$AccT == 20), 'logTimm'])

    retCold[i,2] <- sd(data[[3]][which(data[[3]]$Clone == clone[i] &
                                        data[[3]]$AccT == 20), 'logTimm']) /
      sqrt(length(data[[3]][which(data[[3]]$Clone == clone[i] &
                                   data[[3]]$AccT == 20), 'logTimm'])))

    h576Hot[i,1] <- mean(data[[1]][which(data[[1]]$Clone == clone[i]
                                         & data[[1]]$AccT == 28), 'dA576normTProt'])

    h576Hot[i,2] <- sd(data[[1]][which(data[[1]]$Clone == clone[i] &
                                      data[[1]]$AccT == 28), 'dA576normTProt']) /
      sqrt(length(data[[1]][which(data[[1]]$Clone == clone[i] &
                                   data[[1]]$AccT == 28), 'dA576normTProt'])))

    h576Cold[i,1] <- mean(data[[1]][which(data[[1]]$Clone == clone[i]
                                          & data[[1]]$AccT == 20), 'dA576normTProt'])

    h576Cold[i,2] <- sd(data[[1]][which(data[[1]]$Clone == clone[i] &
                                        data[[1]]$AccT == 20), 'dA576normTProt']) /
      sqrt(length(data[[1]][which(data[[1]]$Clone == clone[i] &
                                   data[[1]]$AccT == 20), 'dA576normTProt'])))

  }

  if(plotIt == TRUE){
    #Time to make the mega-plot
    layout(matrix(c(1,2,3), ncol=1))
    par(mar=c(6,4,0,0))
    #panel a
  }
}
```



```

plot(h576Hot[,1], retHot[,1], pch=21, las=1, xlim=c(0,0.1), ylim=c(3.5, 5.5),
     tck=0.01, bg = 'white',
     ylab= expression(paste('log(T'[imm],')')),
     xlab= expression(paste('H'[576]))))
segments(x0 = h576Hot[,1], y0 = retHot[,1]+retHot[,2], y1 = retHot[,1] - retHot[,2])
segments(x0 = h576Hot[,1]+h576Hot[,2], y0 = retHot[,1], x1 = h576Hot[,1]-h576Hot[,2])
points(h576Hot[,1], retHot[,1], pch = 21, bg='white')

points(h576Cold[,1], retCold[,1], pch = 16)
segments(x0 = h576Cold[,1], y0 = retCold[,1]+retCold[,2],
         y1 = retCold[,1] - retCold[,2])

segments(x0 = h576Cold[,1] + h576Cold[,2], y0 = retCold[,1],
         x1 = h576Cold[,1]-h576Cold[,2])
legend('topright', expression(italic('a')), bty='n', cex=1.25)

#panel b
plot(aht, h576Hot[,1] - h576Cold[,1], pch=21, las=1, tck=0.01, ylim=c(-0.01, 0.1),
     bg = 'white', ylab= expression(paste(Delta, 'H'[576])),
     xlab=expression(paste('AHT_warmest (', ~degree , 'C)')), ...)
abline(lm((h576Hot[,1] - h576Cold[,1]) ~ aht))
legend('topright', expression(italic('b')), bty='n', cex=1.25)

#panel c
plot(aht, h576Hot[,1], pch=21, las=1, tck=0.01, bg = 'white', ylim=c(-0.01, 0.1),
     ylab = expression(paste('H'[576])),
     xlab = expression(paste('AHT_warmest (', ~degree , 'C)')), ...)
segments(x0 = aht, y0 = h576Hot[,1] + h576Hot[,2],
         y1 = h576Hot[,1] - h576Hot[,2])
points(aht, h576Hot[,1], pch = 21, bg='white')

points(aht, h576Cold[,1], pch = 16)
segments(x0 = aht, y0 = h576Cold[,1] + h576Cold[,2],
         y1 = h576Cold[,1] - h576Cold[,2])
legend('topright', expression(italic('c')), bty='n', cex=1.25)
}

if(plotIt == FALSE){
  ret <- rbind(retHot, retCold)
  h576 <- rbind(h576Hot, h576Cold)
  acct <- c(rep(28, nrow(retHot)), rep(20,nrow(retCold)))
  return(list(Timm=ret, h576=h576, AccT=acct))
}
}

getFig3(data, aht)

```

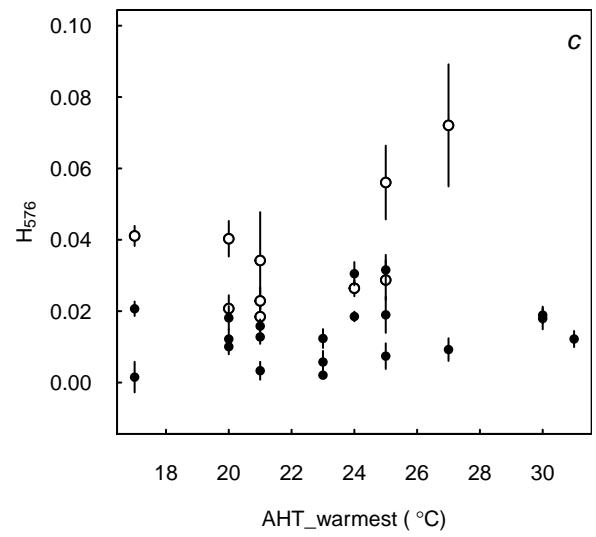
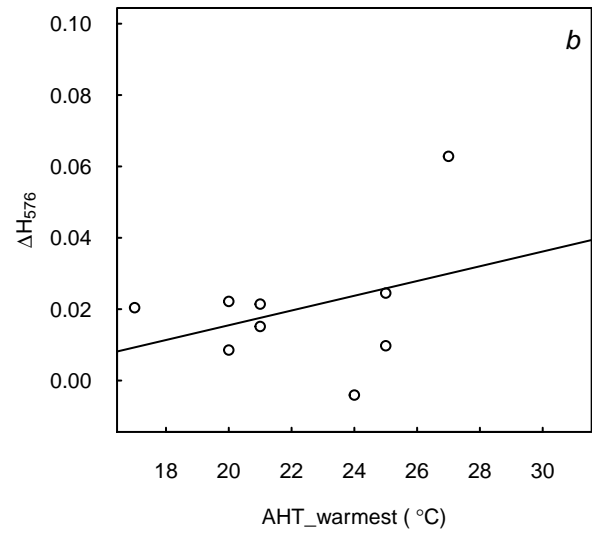
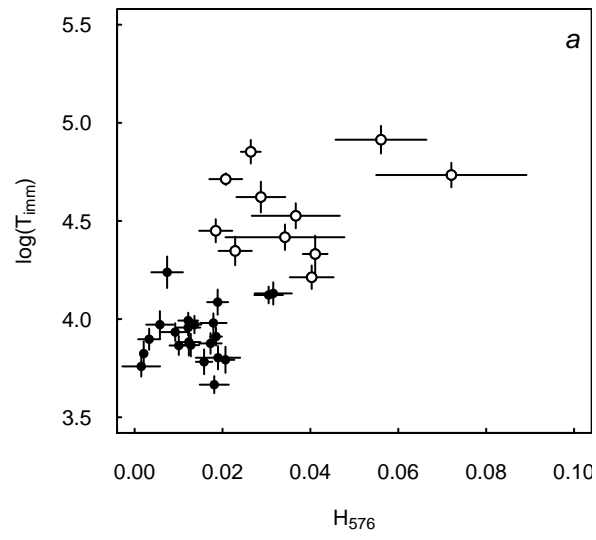


Figure 3:

Table 4

Wrong.

```
getTable4 <- function(data, aht){  
  require(xtable)  
  raw <- getFig3(data, aht, plotIt=FALSE)  
  glmH576 <- aov(glm(log(raw[['h576']][,1]) ~ rep(aht,2) * raw[['AccT']]))  
  glmH576Table <- matrix(unlist(summary(glmH576)), ncol=5)  
  rownames(glmH576Table) <- c("AHT_warmest", "AccT", "Acct * AHT_warmest", "Residuals")  
  colnames(glmH576Table) <- c('df', 'SS', 'MS', 'F', 'p-value')  
  print(xtable(glmH576Table), floating.placement='!', comment=FALSE)  
}  
  
getTable4(data, aht)
```

	df	SS	MS	F	p-value
AHT_warmest	1.00	0.58	0.58	1.15	0.29
AccT	1.00	9.01	9.01	17.94	0.00
Acct * AHT_warmest	1.00	0.00	0.00	0.00	0.96
Residuals	26.00	13.06	0.50		