

Fluctuating temperatures alter environmental pathogen transmission in a *Daphnia*-pathogen system

Tad Dallas and John Drake

This document serves to recreate the figures from

Dallas, T. and J.M. Drake. 2016. “Fluctuating temperatures alter environmental pathogen transmission in a *Daphnia*-pathogen system” *Ecology and Evolution*

Call in libraries

```
library(plyr)
library(dplyr)
library(binom)
```

read data in (individual-level)

```
zero <- read.csv('../Data/individual/indivExpData0.csv')
one <- read.csv('../Data/individual/indivExpData1.csv')
two <- read.csv('../Data/individual/indivExpData2.csv')
four <- read.csv('../Data/individual/indivExpData4.csv')
dat <- dplyr::bind_rows(zero, one, two, four)
```

read data in (spore survival)

```
spSurv <- read.csv('../Data/sporeSurvival/sporeSurv.csv')
```

read data in (filtering rate)

```
balls <- read.csv('../Data/filterRate/filterRate.csv')
justVar <- balls[which(balls$trt %in% c(20, 'v1', 'v2', 'v4')),]
static <- balls[which(balls$trt %in% c(12, 20, 28)),]
```

Support functions

```

add.alpha <- function(col, alpha=1){
  if(missing(col))
    stop("Please provide a vector of colours.")
  apply(sapply(col, col2rgb)/255, 2,
        function(x)
          rgb(x[1], x[2], x[3], alpha=alpha))
}

darken <- function(color, factor=1.4){
  col <- col2rgb(color)
  col <- col/factor
  col <- rgb(t(col), maxColorValue=255)
  col
}

colz <- darken(rev(RColorBrewer::brewer.pal(7, 'Spectral')), 1.25)[c(1,2,5,6,7)]
colz2 <- colz[c(1,2,3,5)]

#How many spores were there in a volume of water
sporeNumber=function(hemoCount, volmicro){
  cellsm1=hemoCount*10000
  cellsmicro=cellsm1/1000
  return(cellsmicro*volmicro)
}

intensity=function(sporemat){
  sp1 <- rowMeans(sporemat[,c('spores1', 'spores2', 'spores3', 'spores4')], na.rm=TRUE)
  vol <- sporemat[, 'volume']
  out <- sporeNumber(sp1, vol)
  out[is.na(out)] <- 0
  return(unlist(out))
}

dat$intensity <- intensity(dat)

```

Create a directory to place all the figures created below.

```
dir.create('figures')
```

Get demographic information

```

getInfo <- function(dat){
  # birth
  totalRepro <- rowSums(dat[,grep('num', colnames(dat))], na.rm=TRUE)
  reproRate <- totalRepro / dat$death.day
  ageAtMaturity <- unlist(dat[, 'Rday1'])
  interBirth <- apply(dat[,grep('Rday', colnames(dat))], 1,
    function(x){mean(diff(na.omit(unlist(x))))})
  interBirth[is.nan(interBirth)] <- 0

  #death

```

```

lifespan <- dat$death.day

#infection
infIntensity <- intensity(dat)

ret <- data.frame(tube=dat$tube, trt=dat$trt, totalRepro,
  reproRate, ageAtMaturity, interBirth, lifespan, infIntensity)
return(ret)
}

```

```

sumData <- getInfo(dat)

```

Infection prevalence (Figure 1)

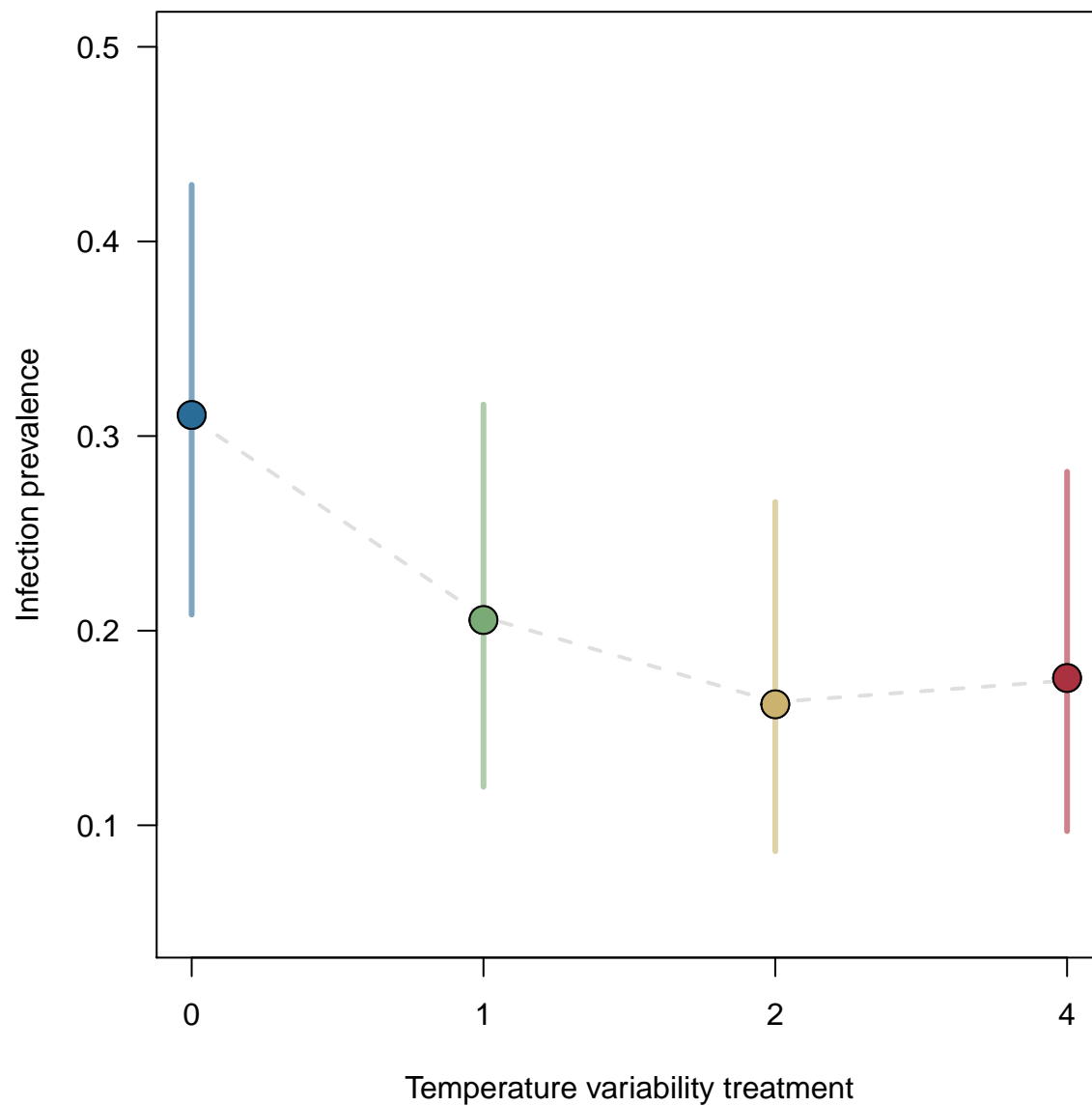
```

prev <- dat %>%
  group_by(trt) %>%
  filter(death.day > 3) %>%
  summarize(numInf= sum(infected,na.rm=TRUE), numAlive = length(infected),
    prev = numInf / length(infected)
  )
dat$infected[is.na(dat$infected)]<-0

prevBars <- binom.confint(prev$numInf, prev$numAlive, methods='exact')

par(mar=c(4,4,0.5,0.5))
plot(c(1:4), prevBars$mean, pch=21, bg=colz[c(1,2,3,5)], col=1, las=1, xaxt='n',
  ylab='Infection prevalence', ylim=c(0.05,0.5), cex=2,
  xlab='Temperature variability treatment')
lines(smooth.spline(dat$infected[which(dat$death.day > 3)] ~
  as.factor(dat$trt[which(dat$death.day > 3)]), spar=0.25),
  col=grey(0.5,0.25),lwd=2, lty=2)
axis(1, labels=c(0,1,2,4), at =1:4)
segments(x0=1:4, y0=prevBars$lower, y1=prevBars$upper,
  col=add.alpha(colz[c(1,2,3,5)], 0.6), lwd=3)
points(x=1:4, prevBars$mean, pch=21, bg=colz[c(1,2,3,5)], col=1, cex=2)

```



```
dev.copy(pdf, 'figures/indPrev.pdf', height=4,width=4);dev.off()
```

```
## pdf
## 3
```

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

Lifespan and infection intensity (Figure 2)

```
intLife <- sumData %>%
  filter(intensity>0)
intLife$intensity <- intLife$intensity / 1000
```

```

layout(matrix(c(1,2,1,3), ncol=2), width=c(1,0.2), height=c(0.2,1))

par(mar=c(0,4,0.5,5))
plot(factor(dat$trt)[-which(dat$intensity==0)],
      dat$death.day[-which(dat$intensity==0)],
      pch=16, las=1, tck=-0.01, ylab='', axes=FALSE,
      xlab='', staplewex=0, notch=F, horizontal=TRUE,
      pars=list(boxwex=0.5, outpch=16, outcol=grey(0.99,0.01), whisklty=1,
      whiskcol=add.alpha(colz[c(1,2,3,5)], 0.8),
      boxcol=add.alpha(colz[c(1,2,3,5)], 0.8),
      staplecol=add.alpha(colz[c(1,2,3,5)], 0.8),
      medcol=add.alpha(colz[c(1,2,3,5)], 0.95)))

points(y=jitter(as.numeric(factor(dat$trt[-which(dat$intensity==0)]))),
       x=dat$death.day[-which(dat$intensity==0)], pch=16, cex=0.7,
       col=add.alpha(colz[1+dat$trt[-which(dat$intensity==0)]], 0.6))

par(mar=c(4,4,0,0))
plot(intLife$lifespan, intLife$infIntensity, pch=16,
      col=add.alpha(colz[1+intLife$trt], 0.99), tck=0.01, las=1,
      ylab='Infection intensity (# spores x1000)', xlab='Time (days)', ylim=c(0,90))

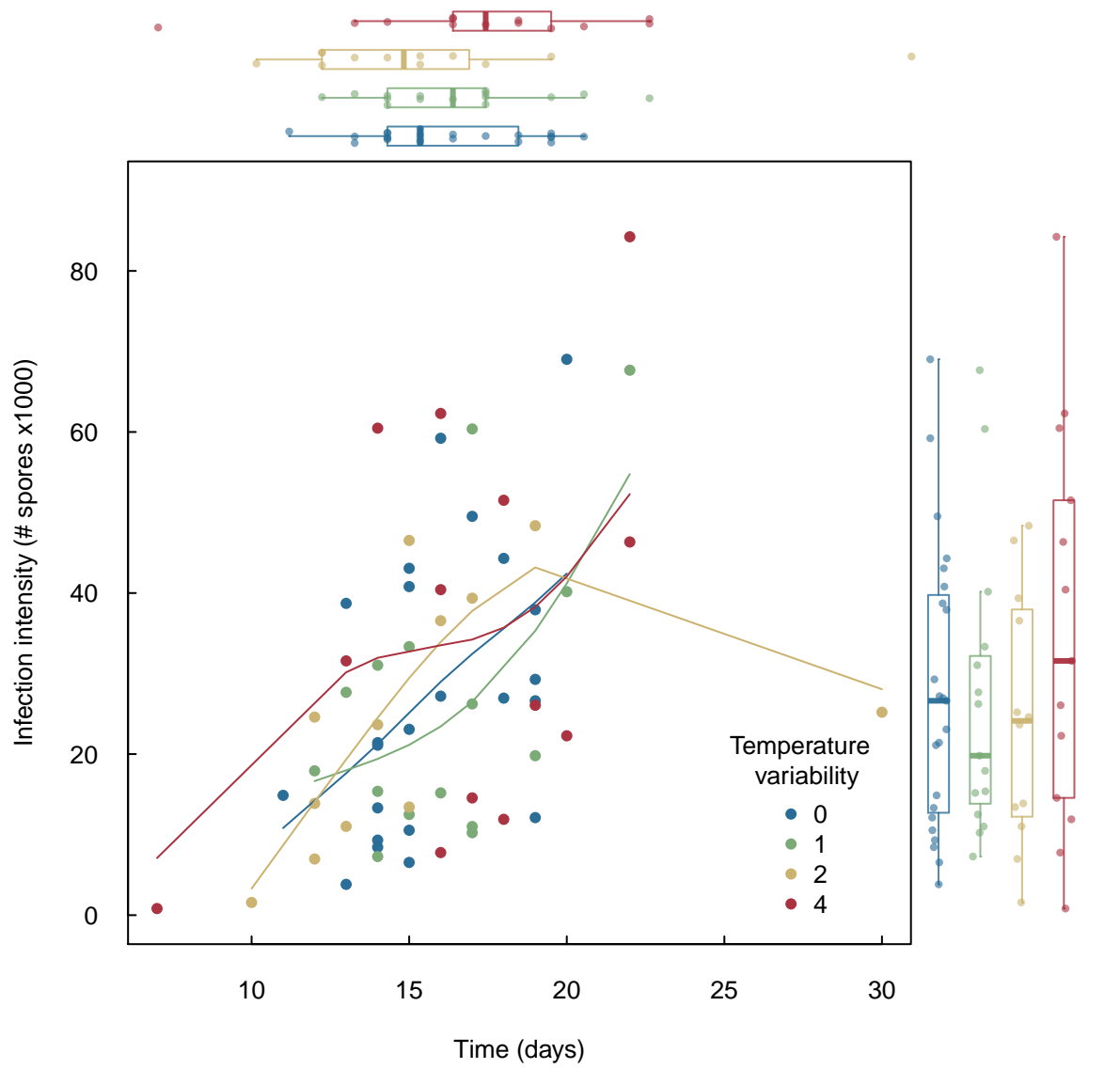
lines(smooth.spline(y=intLife$infIntensity[intLife$trt == 0],
                    x= intLife$lifespan[intLife$trt == 0], spar=0.6), col=colz[1])
lines(smooth.spline(y=intLife$infIntensity[intLife$trt == 1],
                    x= intLife$lifespan[intLife$trt == 1], spar=0.6), col=colz[2])
lines(smooth.spline(y=intLife$infIntensity[intLife$trt == 2],
                    x= intLife$lifespan[intLife$trt == 2], spar=0.6), col=colz[3])
lines(smooth.spline(y=intLife$infIntensity[intLife$trt == 4],
                    x= intLife$lifespan[intLife$trt == 4], spar=0.6), col=colz[5])

legend(25, 20, c(0,1,2,4), title='Temperature \n variability', ncol=1,
      pch=16, col=colz[c(1,2,3,5)], bty='n')

par(mar=c(4,0,0,0))
plot(factor(dat$trt)[-which(dat$intensity==0)],
      dat$intensity[-which(dat$intensity==0)] / 1000,
      pch=16, las=1, tck=-0.01, ylab='', axes=FALSE,
      xlab='', ylim=c(0, 90), staplewex=0, notch=F,
      pars=list(boxwex=0.5, outpch=16, outcol=grey(0.99,0.01), whisklty=1,
      whiskcol=add.alpha(colz[c(1,2,3,5)], 0.8),
      boxcol=add.alpha(colz[c(1,2,3,5)], 0.8),
      staplecol=add.alpha(colz[c(1,2,3,5)], 0.8),
      medcol=add.alpha(colz[c(1,2,3,5)], 0.95)))

points(jitter(as.numeric(factor(dat$trt[-which(dat$intensity==0)]))),
      dat$intensity[-which(dat$intensity==0)] / 1000, pch=16, cex=0.7,
      col=add.alpha(colz[1+dat$trt[-which(dat$intensity==0)]], 0.6))

```



```
dev.copy(pdf, 'figures/IntensityOverTime.pdf', height=5, width=5);dev.off()
```

```
## pdf
## 3
```

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

Spore survival (Figure 3)

```
spSurv2 <- spSurv %>%
  group_by(day, trt) %>%
  summarize(mn = mean(1-(dead/total)),
            se = sd((1-(dead/total))) / sqrt(length(dead)))
```

```

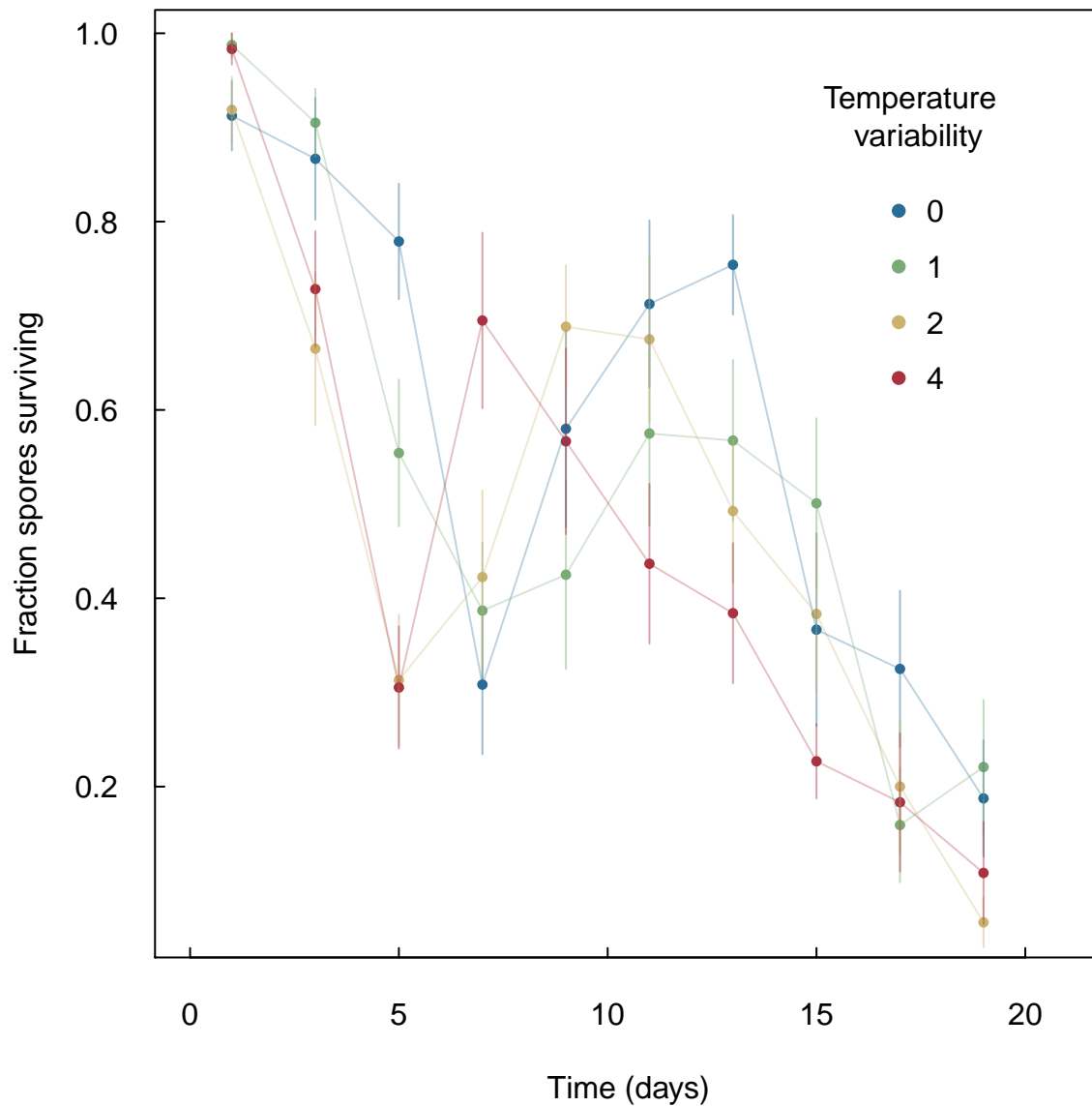
par(mar=c(4,4,0.5,0.5))
plot(spSurv2$day, spSurv2$mn, col=colz2[factor(spSurv2$trt)], las=1, pch=16, cex=0.75,
     xlab='Time (days)', ylab='Fraction spores surviving', tck=0.01, xlim=c(0,21))

segments(x0=spSurv2$day, y0=spSurv2$mn + spSurv2$se, y1=spSurv2$mn - spSurv2$se,
         col=add.alpha(colz2[factor(spSurv2$trt)], 0.45))

tret <- unique(spSurv2$trt)
for(i in 1:length(unique(spSurv2$trt))){
  tmp <- filter(spSurv2, trt == tret[i])
  lines(tmp$day, tmp$mn, col=add.alpha(colz2[i], 0.3))
}

legend(15, 0.92, c(0,1,2,4), title='Temperature \n variability', ncol=1,
      pch=16, col=colz2, bty='n', y.intersp=1.5)

```



```
dev.copy(pdf, 'figures/sporeSurvival.pdf', width=4, height=4); dev.off()
```

```
## pdf
## 3
```

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

Filtering rate plotting (Figure 4)

```
getMeans <- function(data){
  trt <- unique(data$trt)
  mout <- matrix(0, ncol=4, nrow=length(trt))
  seout <- matrix(0, ncol=4, nrow=length(trt))

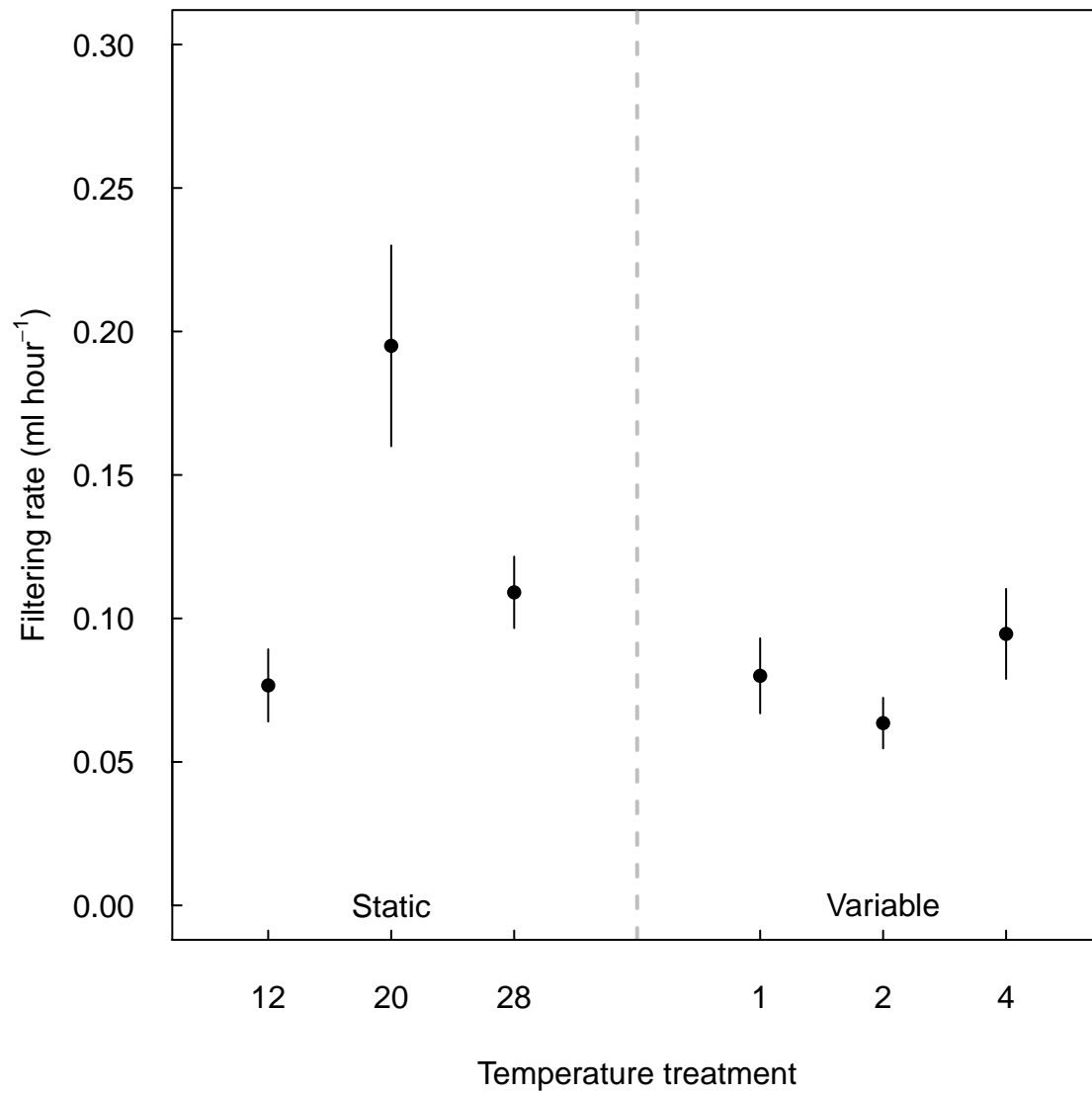
  for(i in 1:length(trt)){
    temp <- data[which(data$trt == trt[i]),]
    mout[i,] <- colMeans(temp[,c('bodyLength', 'Frgut', 'Frbody', 'Fr')])
    seout[i,] <- apply(temp[,c('bodyLength', 'Frgut', 'Frbody', 'Fr')],2,sd) / sqrt(20)
  }
  rownames(mout) <- rownames(seout) <- trt
  colnames(mout) <- colnames(seout) <- c('bodyLength', 'Frgut', 'Frbody', 'Fr')
  mout <- as.data.frame(mout)
  seout <- as.data.frame(seout)
  return(list(means=mout, se=seout))
}
```

```
varMeans <- getMeans(balls)

par(mar=c(5,5,0.5,0.5))
plot(c(1,2,3,5,6,7), varMeans$means[, 'Fr'], xaxt='n', las=1, xlab='',
     ylab='', ylim=c(0,0.3), xlim=c(0.5,7.5), pch=16, tck=0.01)
abline(v=4, lwd=2, lty=2, col=grey(0.5,0.5))

axis(1, at=c(1,2,3,5,6,7), c('12', '20', '28', '1', '2', '4'), tck=0.01)
mtext('Temperature treatment', side=1, line=3)
mtext(expression(paste('Filtering rate (ml hour'-1,'))), side=2, line=3)
text(2,0, 'Static'); text(6,0, 'Variable')

segments(x0=c(1,2,3,5,6,7), y0=(varMeans$means[, 'Fr'])-(varMeans$se[, 'Fr']),
         y1=(varMeans$means[, 'Fr'])+(varMeans$se[, 'Fr']))
```

```
dev.copy(pdf, 'figures/filter.pdf', width=4.4, height=3.6); dev.off()
```

```
## pdf
## 3
```

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

Survival analysis (Figure S1)

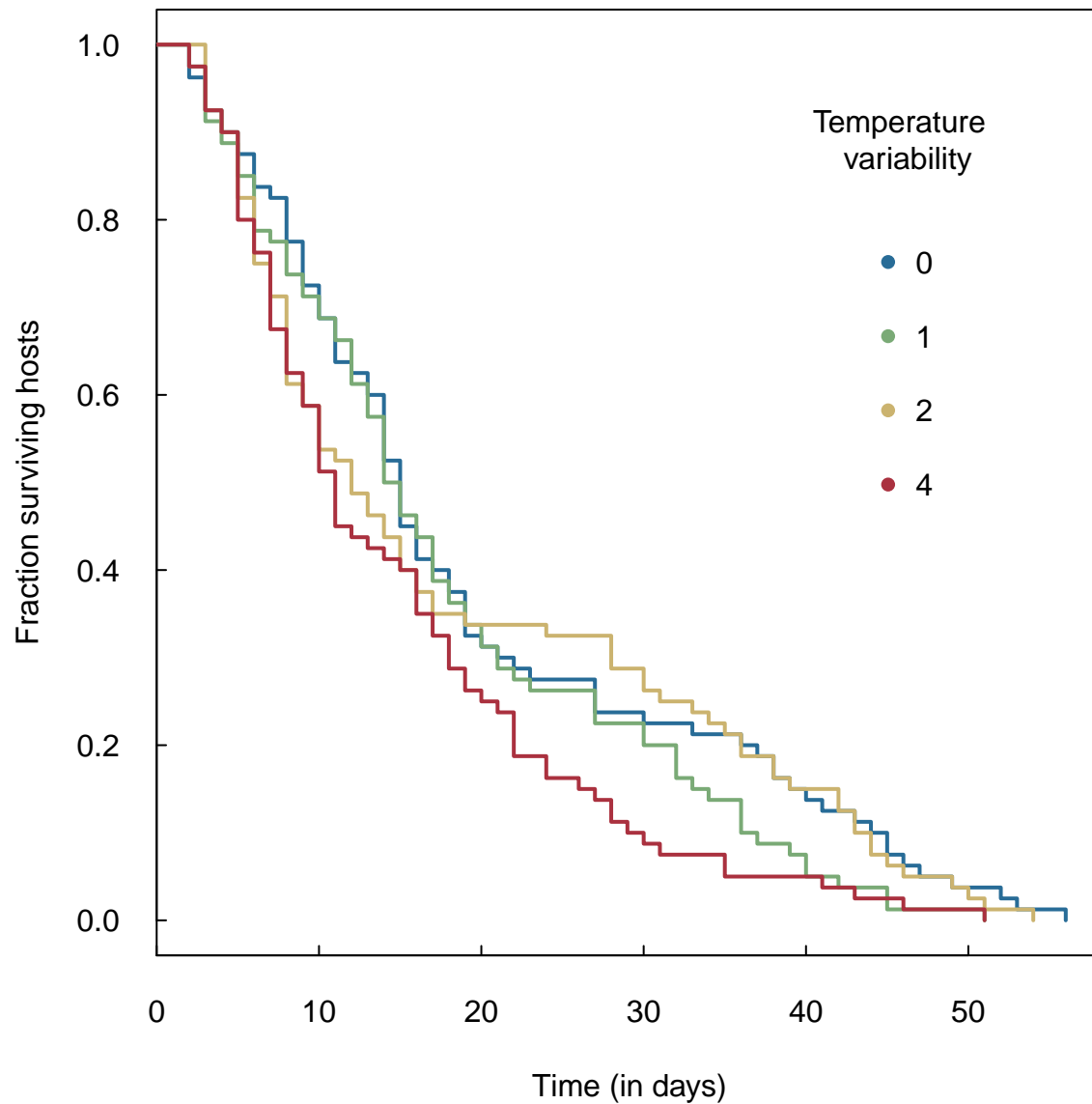
datCox corresponds to Table 1 in the manuscript.

```
library(survival)
dat$infected[is.na(dat$infected)] <- 0
datSurv <- Surv(dat$death.day)
datCox <- coxph(datSurv ~ as.factor(trt) + infected, data=dat)
datFit <- survfit(datSurv ~ dat$trt)
```

```

par(mar=c(4,4,0.5,0.5))
plot(datFit, col=colz[c(1,2,3,5)], conf.int=FALSE, mark.time=TRUE, lwd=2,
     las=1, tck=0.01, ylab='Fraction surviving hosts', xlab='Time (in days)')
legend(40, 0.9, c(0,1,2,4), title='Temperature \n variability', ncol=1,
     pch=16, col=colz[c(1,2,3,5)], bty='n', y.intersp=2)

```



```

dev.copy(pdf, 'figures/survival.pdf', height=3.75, width=3.75);dev.off()

```

```

## pdf
## 3

```

```

## pdf
## 2

```

Filtering rate supplemental plot (Figure S2)

```

varMeans <- getMeans(balls[which(balls$trt %in% c(20,'v1','v2','v4')),])
varMeans$means$Frgut <- (varMeans$means$Frgut / varMeans$means$Fr)
varMeans$means$Frbody <- (varMeans$means$Frbody / varMeans$means$Fr)

par(mar=c(5,5,0.5,0.5))
plot(c(1,2,3,4), 20*varMeans$means$Fr, xaxt='n', las=1, xlab='', ylab='',
      xlim=c(0.5,4.5), pch=16, cex=0.1, tck=0.01, ylim=c(1, 5), yaxt='n')

axis(1, at=c(1,2,3,4), c('0', '1', '2', '4'), tck=0.01)
axis(2, at=c(1,2,3,4,5), labels=(1:5)/20, las=1, tck=0.01)
mtext('Temperature variability treatment', side=1, line=3)
mtext(expression(paste('Filtering rate (ml hour'~{-1},')')), side=2, line=3)

segments(x0=c(1,2,3,4), y0=20*(varMeans$means$Fr -varMeans$se$Fr),
          y1= 20*(varMeans$means$Fr + varMeans$se$Fr), lwd=4, col=grey(0.1,0.5))

Rgraphviz::pieGlyph(x=as.numeric(c(varMeans$means[1, 2:3])), xpos=1,
                    ypos=20*varMeans$means$Fr[1], radius=0.15, col=c('#FF8D1E','dodgerblue'),
                    border=darken(c('#FF8D1E','dodgerblue')), labels=c('',''))

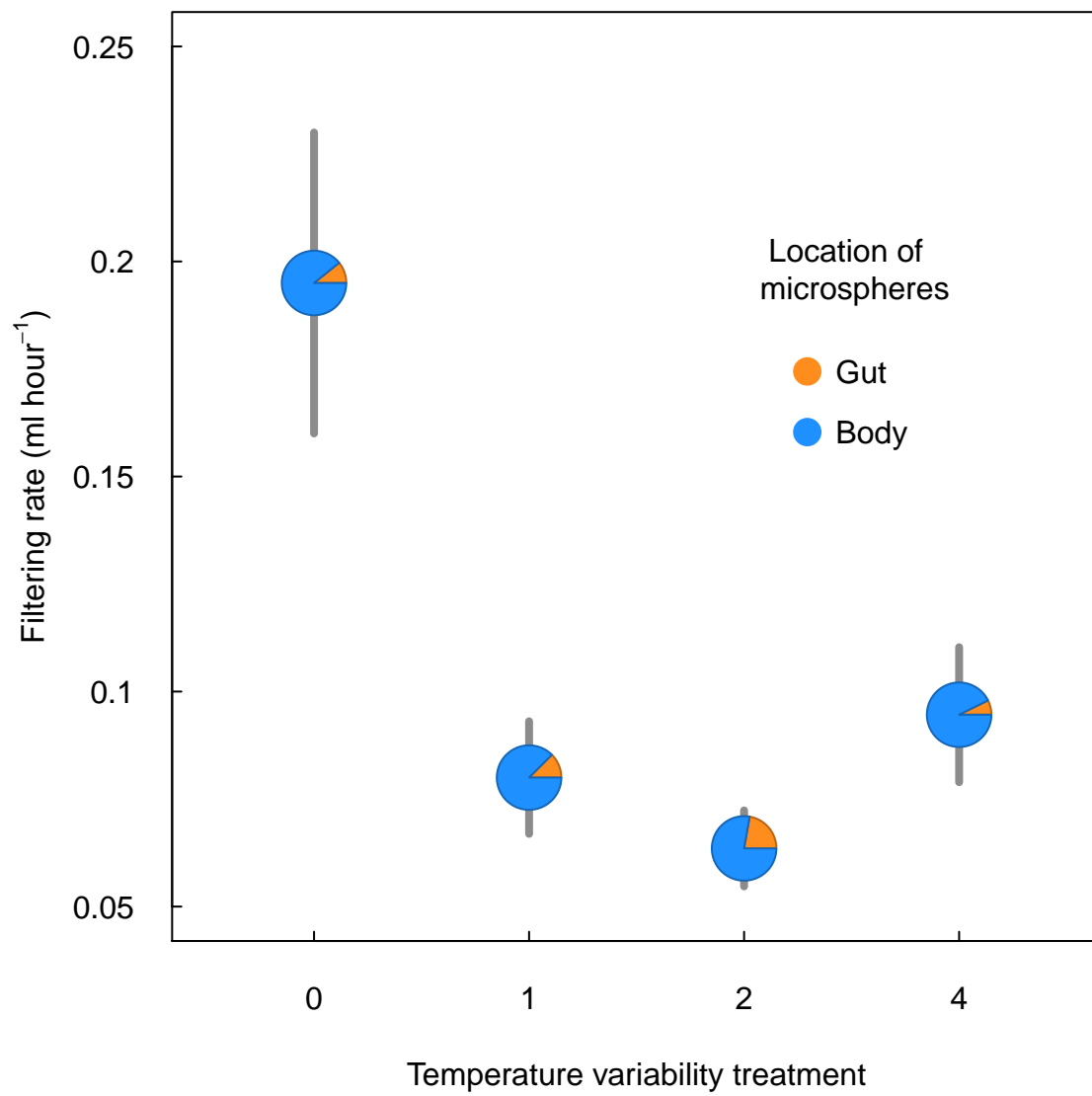
Rgraphviz::pieGlyph(x=as.numeric(c(varMeans$means[2, 2:3])), xpos=2,
                    ypos=20*varMeans$means$Fr[2], radius=0.15, col=c('#FF8D1E','dodgerblue'),
                    border=darken(c('#FF8D1E','dodgerblue')), labels=c('',''))

Rgraphviz::pieGlyph(x=as.numeric(c(varMeans$means[3, 2:3])), xpos=3,
                    ypos=20*varMeans$means$Fr[3], radius=0.15, col=c('#FF8D1E','dodgerblue'),
                    border=darken(c('#FF8D1E','dodgerblue')), labels=c('',''))

Rgraphviz::pieGlyph(x=as.numeric(c(varMeans$means[4, 2:3])), xpos=4,
                    ypos=20*varMeans$means$Fr[4], radius=0.15, col=c('#FF8D1E','dodgerblue'),
                    border=darken(c('#FF8D1E','dodgerblue')), labels=c('',''))

legend(3,4, pch=16, col=c('#FF8D1E','dodgerblue'),
       legend=c('Gut', 'Body'), title='Location of \n microspheres',
       bty='n', pt.cex=2, y.intersp=1.6)

```



```
dev.copy(pdf, 'figures/supFilter.pdf', width=4.4, height=4.4); dev.off()
```

```
## pdf
## 3
```

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