

Competition-mediated feedbacks in experimental multi-species epizootics

Tad Dallas, Richard Hall, John Drake

```
#Load some data in
load('Dallas2016.RData')

#load needed packages
#library(reshape2);
library(lattice);
library(deSolve);
library(dplyr)

# devtools::install_github('karthik/wesanderson')
library(wesanderson)
```

Dallas, T, R Hall, and JM Drake. 2015. Competition-mediated feedbacks in experimental multi-species epizootics. Ecology. <http://dx.doi.org/10.1890/15-0305.1>

The SIPR model

```
# Filtering rate functions
frS <- function(R){ return(0.02 / (1+4*R))}

frC <- function(R){ return(0.025/(1+4*R))}

SIP=function(t,y,parms){
  require(deSolve)
  Sdt <- y[1]; #Susceptibles
  Idt <- y[2]; #Infected individuals
  Pdt <- y[3]; #Pathogen
  Rdt <- y[4]; #Resources
  Cdt <- y[5]; #Competitor

  #Model parameters
  phi=parms[1]; muS=parms[2]; muC=parms[3]; muI=parms[4]; muP=parms[5];
  muR=parms[6]; eS=parms[7]; eC=parms[8]; u=parms[9]; theta=parms[10];
  zS=parms[11]; zC=parms[12]; f=parms[13]

  dSdt= (eS*frS(Rdt)*Rdt*(Sdt+(Idt*f))) - (muS*Sdt) - (u*frS(Rdt)*Sdt*Pdt);

  dIdt= (u*frS(Rdt)*Sdt*Pdt)- (muI*Idt);

  dPdt= (theta*muI*Idt) - (muP*Pdt) - (zS*frS(Rdt)*(Sdt+Idt)*Pdt) - (zC*frC(Rdt)*Cdt*Pdt);

  dRdt= phi - (muR*Rdt)- (frS(Rdt)*Rdt*(Sdt+Idt)) - (frC(Rdt)*Rdt*Cdt)

  dCdt= (eC*frC(Rdt)*Rdt*Cdt) - (muC*Cdt)
```

```

    return(list(c(dSdt, dIdt, dPdt, dRdt, dCdt)))
}

```

The dimensionless SIPR model

```

sip=function(t,y,parms){
  sdt=y[1]; #Susceptibles
  idt=y[2]; #Infected individuals
  pdt=y[3]; #Pathogen
  rdt=y[4]; #Resources
  cdt=y[5]; #Competitor

  #Model parameters
  muRS=parms[1]; muIS=parms[2]; muCS=parms[3]; muPS=parms[4]; phi=parms[5];
  z=parms[6]; zetaCS=parms[7]; a=parms[8]; alphaCS=parms[9]; g=parms[10];
  gammaCS=parms[11]; w=parms[12]

  # update equations to dimensionless

  dsdt= ((g/(1+a*rdt))*rdt*(sdt+(idt*phi))) - (sdt) - ((1/(1+a*rdt))*sdt*pdt);

  didt= ((1/(1+a*rdt))*sdt*pdt)- (muIS*idt);

  dpdt= (w*muIS*idt) - (muPS*pdt) - ((z/(1+a*rdt))*(sdt+idt)*pdt) -
    ((zetaCS*z*muCS/(1+alphaCS*a*rdt))*cdt*pdt);

  drdt= muRS*(1-rdt) - ((1/(1+a*rdt))*rdt*(sdt+idt)) -
    ((muCS/(1+alphaCS*a*rdt))*rdt*cdt)

  dcdt= ((gammaCS*g/(1+alphaCS*a*rdt))*rdt*cdt) - (muCS*cdt)

  return(list(c(dsdt, didt, dpdt, drdt, dcdt)))
}

```

Parameter estimates

```

# Dimensional initial conditions (fixed)
## Assume dynamics are occurring in 1 liter
## Measures are based on timestep of 1 day

Sdt0 = 30 ; #initial number of uninfected hosts
Idt0 = 0 ; #initial number of infected individuals
Pat0 = 10000 ; #initial condition for pathogen (10 spores per ml)
Res0 = 0.67 ; #mg dry weight per liter
Cdt0 = 0
Cmin0 = 0
Cmax0 = 100

# first list dimensional params needed to rescale

```

```

pie=0.67; muS=0.1; muC=0.1; muI=0.15; muP=0.25; muR=0.25;
eS=25; eC=28; u=2.03e-4; theta=20000; zS=0.30; zC=0.30;
f=0.75; fS0=0.02 ; fC0=0.025; fS1=4; fC1=4

parmsDim = c(pie, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)

# Create a nice two color gradient
# (named `tom` after co-creator Tomlin Pulliam)
tom <- colorRampPalette(c("#FFFFFF", "#2D4A81"))

dir.create('Figures')

```

```
## Warning in dir.create("Figures"): 'Figures' already exists
```

Manuscript plots

Figure 1

```

comp=seq(0,100, length.out=100)
prevLowR = vector()
prevMidR = vector()
prevHighR = vector()
parmsDim=c(0.5, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)
parmsDim2=c(0.67, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)
parmsDim3=c(1, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)

for(i in 1:length(comp)){
  outDim=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim)
  outDim2=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim2)
  outDim3=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim3)
  prevLowR[i] = mean(outDim[,3]/(outDim[,2]+outDim[,3]))
  prevMidR[i] = mean(outDim2[,3]/(outDim2[,2]+outDim2[,3]))
  prevHighR[i] = mean(outDim3[,3]/(outDim3[,2]+outDim3[,3]))
}

par(mar=c(4,4,0.5,1))
plot(prevLowR, ylim=c(0,0.1), las=1, ylab='Infection prevalence',
      xlab=expression(paste("Initial competitor density ( $L^{-1}$ ,"))),
      type='l', col=grey(0.5), tck=0.01, xlim=c(0,105), lty=3, lwd=2)
lines(prevMidR, lwd=3, col=1)
lines(prevHighR, lwd=2, col=grey(0.5), lty=2)

legend(60, 0.09, c(expression(paste(pi, " = 0.5")), expression(paste(pi, " = 0.67")),
                    expression(paste(pi, " = 1.0"))), lty=c(3,1,2),lwd=2,
      col=c(grey(0.5), 1, grey(0.5)), bty='n', cex=1, y.intersp = 2)

dev.copy(pdf, 'Figures/prev2.pdf', height=4.25, width=4.25);dev.off()

```

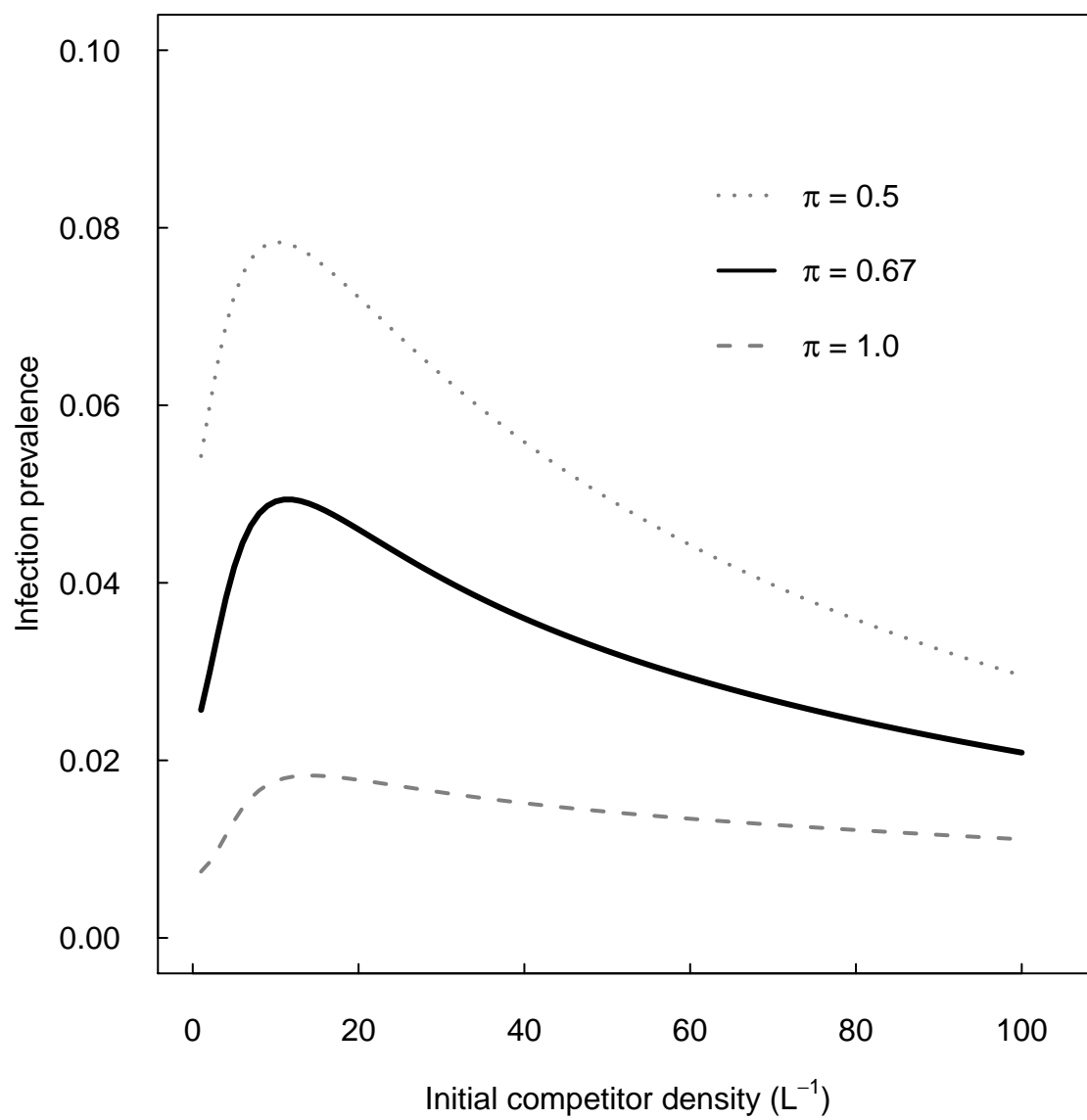


Figure 1:

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

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

Figure 2

```
comp <- seq(0,100, length.out=100)
susLowR <- vector()
susMidR <- vector()
susHighR <- vector()

parmsDim <- c(0.5, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)
parmsDim2 <- c(0.67, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)
parmsDim3 <-c(1, muS, muC, muI,muP, muR, eS, eC, u, theta, zS, zC, f)

for(i in 1:length(comp)){
  outDim=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim)
  outDim2=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim2)
  outDim3=lsoda(y=c(Sdt0,Idt0,Pat0,Res0,comp[i]), times=seq(1,70,by=1), func=SIP, parmsDim3)
  susLowR[i] = mean(outDim[,2]+outDim[,3])
  susMidR[i] = mean(outDim2[,2]+outDim2[,3])
  susHighR[i] = mean(outDim3[,2]+outDim3[,3])
}

par(mar=c(4,4.5,0.5,1))
plot(susLowR, las=1, ylab=expression(paste("Mean susceptible density ( $L^{-1}$ ,"))),
      xlab=expression(paste("Initial competitor density ( $L^{-1}$ ,"))), type='l',
      col=grey(0.5), tck=0.01, xlim=c(0,105), lty=3, lwd=2, ylim=c(10, 55))
lines(susMidR, lwd=2, col=1)
lines(susHighR, lwd=2, col=grey(0.5), lty=2)

legend(60, 45, c(expression(paste(pi, " = 0.5")), expression(paste(pi, " = 0.67")),
                  expression(paste(pi, " = 1.0"))), lty=c(3,1,2),lwd=2,
      col=c(grey(0.5), 1, grey(0.5)), bty='n', cex=1, y.intersp = 2)

dev.copy(pdf, 'Figures/demography2.pdf', height=4.25, width=4.25);dev.off()
```

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

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

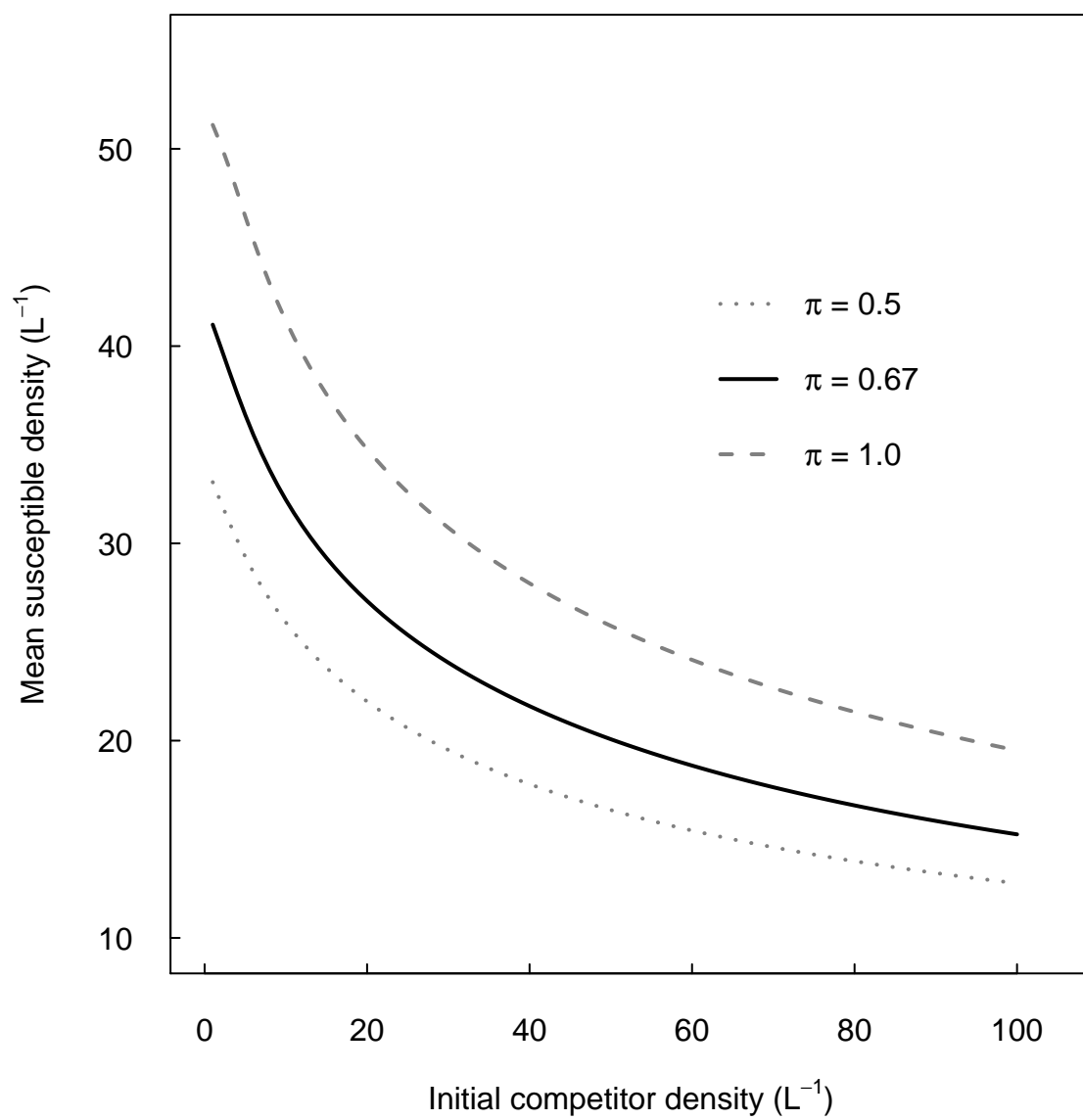


Figure 2:

Experimental epidemics

Below, you'll find data from the experimental epidemics performed, and the code to reproduce the analyses.

```
#Obtains a series of mean values given the data that are epidemic time series
getMeans=function(data){
  popID <- unique(data$id)
  out=matrix(0, ncol=6, nrow=length(popID))
  for(i in 1:length(popID)){
    temp=data[which(data$id == popID[i]),]
    out[i,1]=temp$dp_number[1];

    if(any(temp$pulic_density == 0)){
      out[i,2]=mean(temp$pulic_density[-which(temp$pulic_density == 0)]);
    }else{
      out[i,2]=mean(temp$pulic_density)
    }
    out[i,3]=mean(temp$dent_density[-which(temp$dent_density==0)]);
    out[i,4]=mean(temp$infection_prev[-which(temp$dent_density==0)]);

    out[i,5]=temp$day[which(temp$dent_density==0)[1]];
    a <- temp[which(temp$infection_prev>0),];
    out[i,6]= max(a$day)-min(a$day)
  }
  colnames(out) <- c('trt','meanPulic', 'meanDenti', 'meanPrev', 'extDenti', 'epiDuration')
  rownames(out)=popID
  return(out)
}

meansAll <- as.data.frame(getMeans(data))
```

Figure 4

```
layout(matrix(c(1,2,3), byrow=TRUE,ncol=1), heights=c(1,1,1.1))
m <- c(0,30,100)
par(mar=c(2, 4.45,0.5,0.5))
#mean prevalence
mprev <- meansAll %>%
  group_by(trt) %>%
  summarize(mn = mean(meanPrev), se = sd(meanPrev)/sqrt(5))

plot(m, mprev[['mn']], pch=16, ylim=c(0,0.12), xlim=c(0,110), xlab='', ylab='',
      tck=0.01,las=1, xaxt='n', cex=1.5)
mtext("Mean prevalence", side=2, line=3.35, cex=0.75)
segments(x0=m, y0= mprev[['mn']] - mprev[['se']], x1=m, y1= mprev[['mn']] + mprev[['se']])
text(35, 0.09, '*', cex=2)
axis(1, at=m, labels=c('', '', ''), tck=0.025)

ext <- meansAll %>%
  group_by(trt) %>%
```

```

        summarize(mn = mean(extDenti), se = sd(extDenti)/sqrt(5))
plot(m, ext[['mn']], pch=16, xlab='', ylab='', tck=0.01, las=1, ylim=c(20,70), tck=0.01,
     xaxt='n', cex=1.5)
segments(x0=m, y0=ext[['mn']] + ext[['se']], y1=ext[['mn']] - ext[['se']])
mtext("Time until extinction (days)", side=2, line=2.5, cex=0.75)
axis(1, at=m, labels=c('', '', ''), tck=0.025)

dur <- meansAll %>%
  group_by(trt) %>%
  summarize(mn = mean(epiDuration), se = sd(epiDuration)/sqrt(5))
par(mar=c(4, 4.45, 0.5, 0.5))
plot(m, dur[['mn']], pch=16, ylim=c(0,45), xlim=c(0,110), xlab='', ylab='', tck=0.01, las=1,
     xaxt='n', cex=1.5)
#mtext(lab, side=1, line=2.5)
text(34.5, 35, '*', cex=2)
mtext("Epidemic duration", side=2, line=2.35, cex=0.75)
segments(x0=m, y0= dur[['mn']] - dur[['se']], x1=m, y1=dur[['mn']] + dur[['se']])
axis(1, at=m, labels=m, tck=0.025)

lab <- expression(paste("Initial competitor density ( $L^{-1}$ ),") (C"[0],"))
mtext(lab, side=1, line=2.75, cex=0.75)

dev.copy(pdf, 'Figures/experiment.pdf', height=8.5, width=3.5); dev.off()

```

```

## pdf
## 3

```

```

## pdf
## 2

```

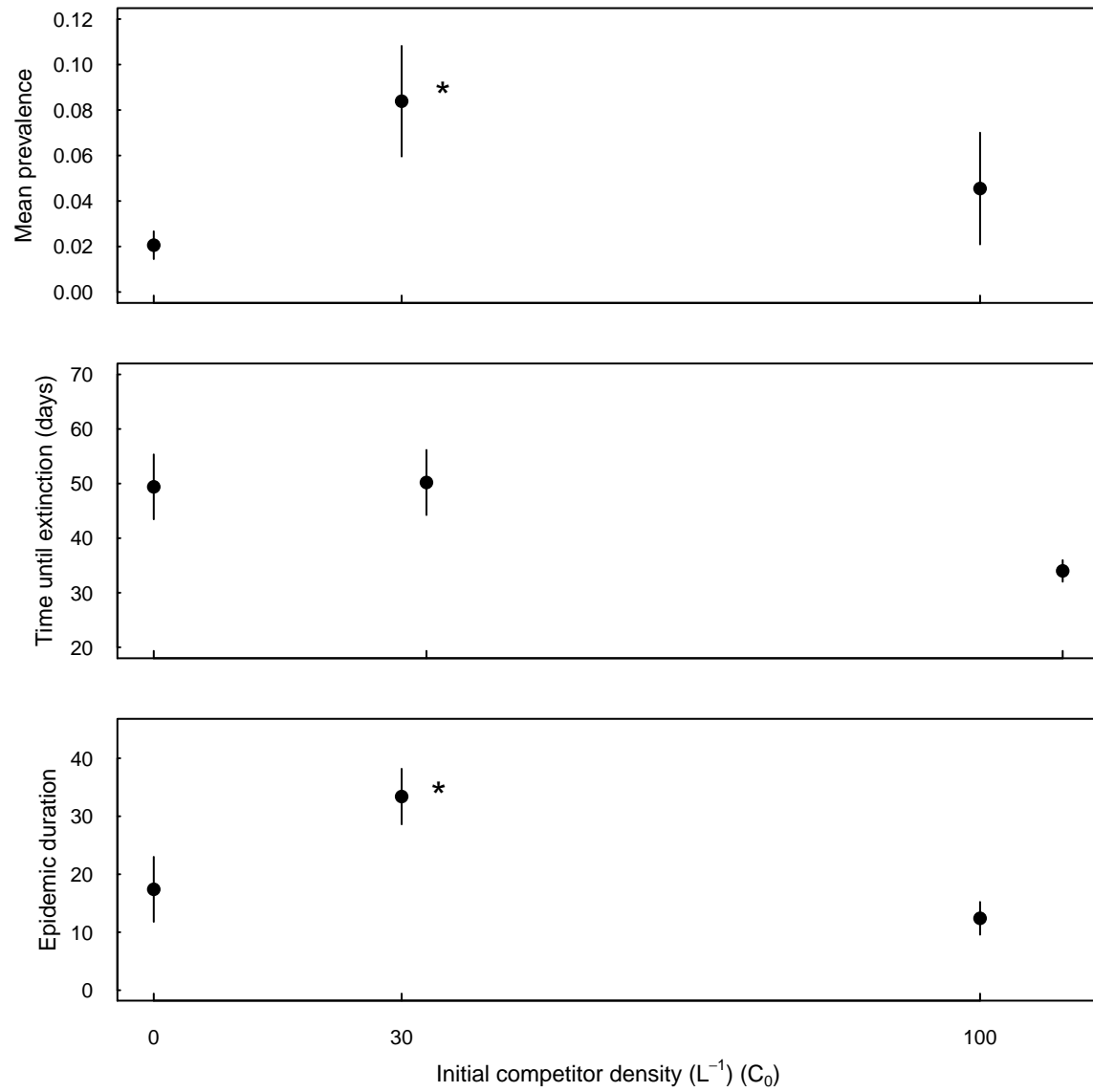



Figure 3:

Sensitivity Analysis

```

getSensitive = function(sensitiveParmIndex, range=NULL, compN=100, path=10000){

  Sdt0 <- 30      ; #initial number of uninfected hosts
  Idt0 <- 0       ; #initial number of infected individuals
  if(is.null(path)){Pat0 <- 10000}else{Pat0 <- path} ; #initial condition for pathogen (10 spores per m
  Res0 <- 0.67    ; #mg dry weight per liter
  Cdt0 <- 0
  Cmin0 <- 0
  Cmax0 <- 100

  # first list dimensional params needed to rescale
  pie <- 0.67; muS <- 0.1; muC <- 0.1; muI <- 0.15; muP <- 0.25;
  muR <- 0.25; eS <- 26; eC <- 28; u <- 2.03e-4; theta <- 20000;
  zS <- 0.30; zC <- 0.30; f <- 0.75; fS0 <- 0.02 ; fC0 <- 0.025;
  fS1 <- 4; fC1 <- 4

  # calculate dimensionless ICs
  sdt0 <- Sdt0*fS0/muS
  idt0 <- 0
  pat0 <- Pat0*u*fS0/muS
  res0 <- Res0*muR/pie
  cdt0 <- 0
  cmin0 <- Cmin0*fC0/muC
  cmax0 <- Cmax0*fC0/muC
  cRange <- seq(cmin0, cmax0, length.out=compN)

  Y0 <- c(sdt0, idt0, pat0, res0, cdt0)

  # values for dimensionless model parameters
  muRS=muR/muS; muIS=muI/muS; muCS=muC/muS; muPS=muP/muS; phi=f;
  z=zS; zetaCS=zC/zS; a=(fS1*pie)/muR ; alphaCS=fC1/fS1; g = (eS*fS0*pie)/(muS*muR);
  gammaCS=(eC*fC0)/(eS*fS0); w=u*theta

  parms=c(muRS, muIS, muCS, muPS, phi, z, zetaCS, a, alphaCS, g, gammaCS, w)

  # Experiment was run for 70 days. using muS = 0.1,
# this means we follow pathogen dynamics for 7 generations
  timevec=seq(0.1,7,by=0.1)

  ret = matrix(0, ncol=length(range), nrow = compN)

  for(i in 1:length(range)){
    parms[sensitiveParmIndex] = range[i]
    for(j in 1:length(cRange)){
      Y0[5] = cRange[j]
      output=lsoda(y=Y0,times=timevec, func=sip, parms)
      ret[j,i] = mean(output[,3]/(output[,2] + output[,3]))
    }
  }
  return(ret)
}

```

```

getSensitiveDim = function(sensitiveParmIndex, range=NULL, compN=100){
  Sdt0 = 30 ; #initial number of uninfected hosts
  Idt0 = 0 ; #initial number of infected individuals
  Pat0 = 10000 ; #initial condition for pathogen (10 spores per ml)
  Res0 = 0.67 ; #mg dry weight per liter
  Cdt0 = 0
  cRange=seq(0,100,length.out=compN)
  Y0 = c(Sdt0, Idt0, Pat0, Res0, Cdt0)
  parameters=c(phi=0.67, muS=0.1, muC=0.1, muI=0.15, muP=0.25, muR=0.25,
               eS=26, eC=28, u=2.03e-4, theta=20000, zS=0.30, zC=0.30, f=0.75)

  timevec=seq(1,70,by=1)
  ret = matrix(0, ncol=length(range), nrow = compN)

  for(i in 1:length(range)){
    parameters[sensitiveParmIndex] = range[i]
    for(j in 1:length(cRange)){
      Y0[5] = cRange[j]
      output=lsoda(y=Y0,times=timevec, func=SIP, parameters)
      ret[j,i] = mean(output[,3]/(output[,2] + output[,3]))
    }
  }
  return(ret)
}

```

```

cols <- wes_palette("Darjeeling")

makePlot = function(SensitiveObject, col=cols[1], ...){
  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))
  }
  plot(SensitiveObject[,50], las=1, tck=0.01, lwd=0, type='n', ...)
  quants = apply(SensitiveObject, 1, quantile, probs=c(0, 1))
  polygon(x=c(1:nrow(SensitiveObject), nrow(SensitiveObject):1),
          c(quants[1,], rev(quants[2,])), col=add.alpha(col, alpha=0.5))
}

dir.create('SupplementalFigures')

```

```

## Warning in dir.create("SupplementalFigures"): 'SupplementalFigures'
## already exists

```

Fecundity reduction by infection (ϕ)

```
phiSensitive <- getSensitive(5, seq(0.25,0.9, length.out=100), compN=100)
phiSensitiveH <- getSensitive(5, seq(0.25,0.9, length.out=100), compN=100, path=50000)
phiSensitiveL <- getSensitive(5, seq(0.25,0.9, length.out=100), compN=100, path=5000)
dimPhi <- getSensitiveDim(13, seq(0.25,0.9, length.out=50), compN=100)
```

```
layout(matrix(c(1,2,3),ncol=3))

par(mar=c(4,4,1.5,0.5))
makePlot(phiSensitiveL, ylim=c(0.01,0.17), xlab='', ylab='Infection prevalence',
          main=expression(paste(phi," ; P = 500")))

makePlot(phiSensitive, ylim=c(0.01,0.17), main=expression(paste(phi," ; P = 10000")),
          xlab=expression(paste("Initial competitor density (L"^{-1}),")), ylab='')

makePlot(phiSensitiveH, ylim=c(0.01,0.17), main=expression(paste(phi," ; P = 50000")),
          ylab='', xlab='')
```

```
dev.copy(pdf, 'SupplementalFigures/phiPath.pdf', height=3,width=9);dev.off()
```

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

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

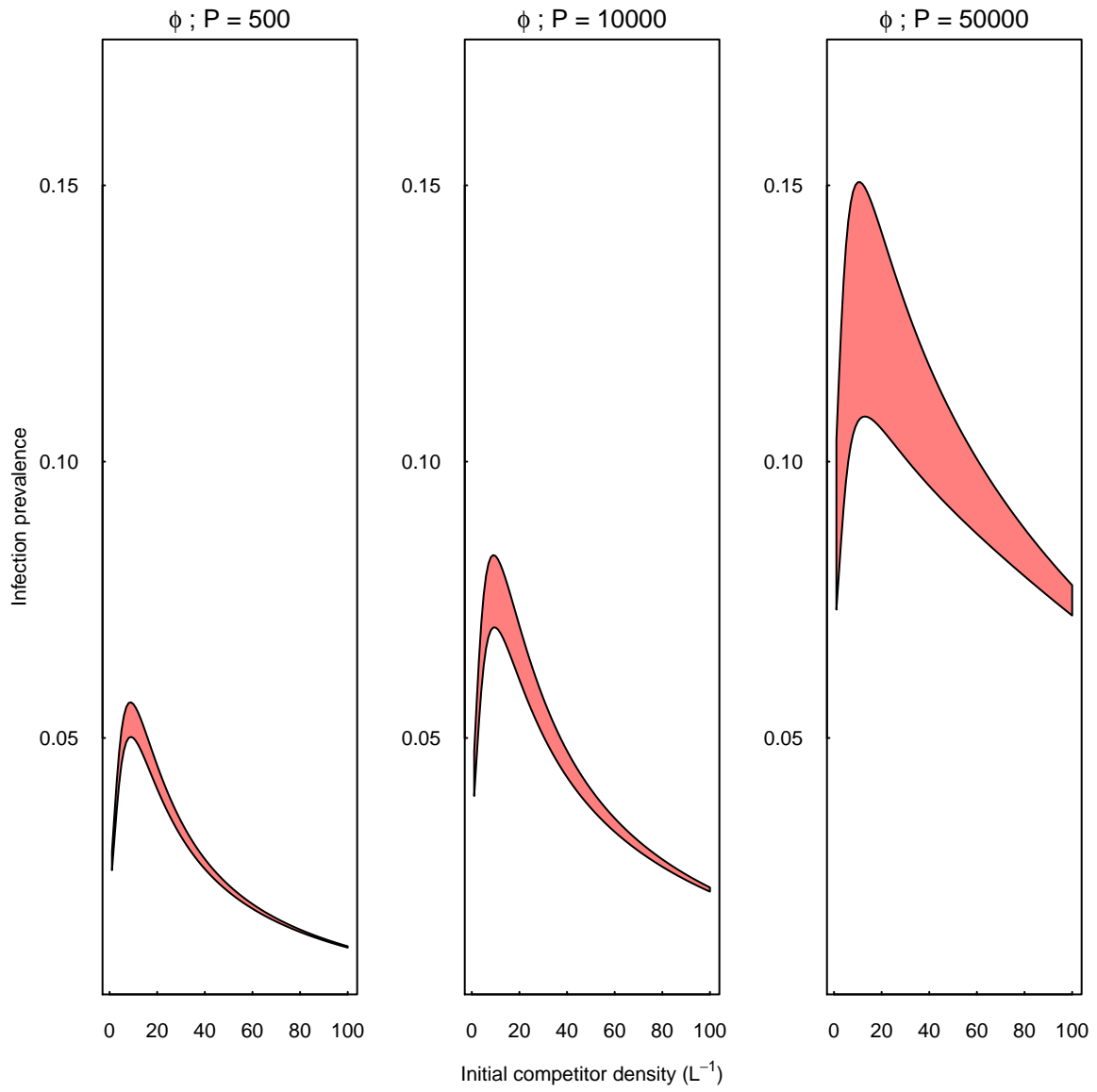


Figure 4:

```

par(mar=c(3,4,1.5,0.5))

makePlot(phiSensitive, col=cols[5], ylim=c(0.01,0.1), xlab='', ylab='Infection prevalence',
        main=expression(phi), xlim=c(0,105))
mtext(expression(paste("Initial competitor density ( $L^{-1}$ )")), side=1, line=2)

lines(phiSensitive[,67], lwd=2, col=1)
arrows(x0=104, y0 = (0.75* min(phiSensitive[nrow(phiSensitive),])),
       y1=(1.25 * max(phiSensitive[nrow(phiSensitive),])))
text(105.75, (0.9 * mean(phiSensitive[nrow(phiSensitive),])), expression(phi), cex=1.5)

```

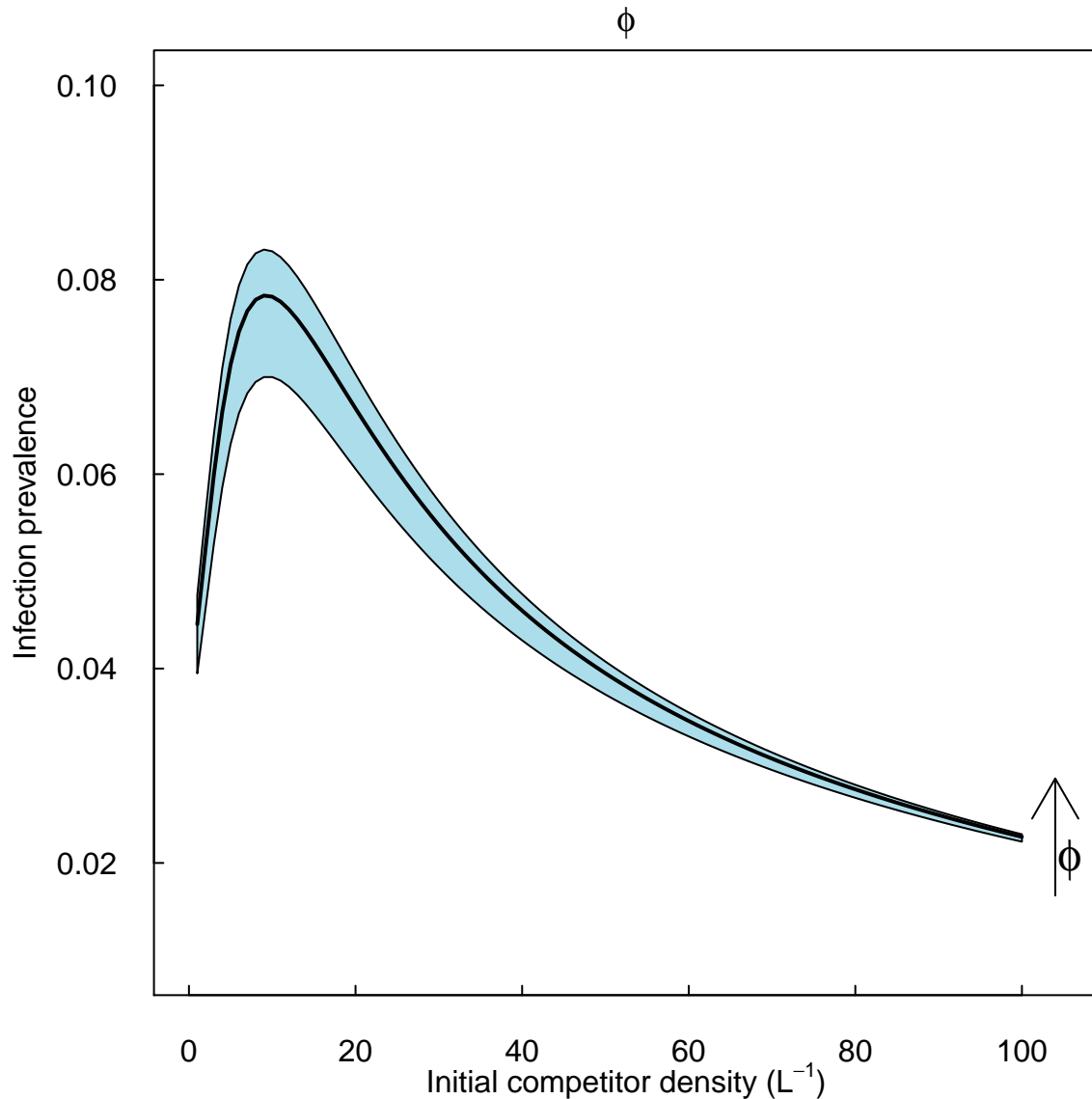


Figure 5:

```

dev.copy(pdf, 'SupplementalFigures/phi.pdf', height=6,width=6);dev.off()

```

```
## pdf
```

```
## 3
```

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

Fraction of spores digested (z)

```
zSensitive <- getSensitive(6, seq(0,0.6, length.out=100), compN=100)
```

```
par(mar=c(3,4,1.5,0.5))  
makePlot(zSensitive, col=cols[2], xlab='', ylab='Infection prevalence',  
          main='z', ylim=c(0,0.35),xlim=c(0,105))  
mtext(expression(paste("Initial competitor density ( $L^{-1}$ ,"))), side=1, line=2)  
  
lines(zSensitive[,50], lwd=2, col=1)  
arrows(x0=104, y1 = (1.25 * (0.001+min(zSensitive[nrow(zSensitive),])),  
        y0=(0.65 * max(zSensitive[nrow(zSensitive),]))  
text(105.5, (1.5 * mean(zSensitive[nrow(zSensitive),])), "z", cex=1.5)
```

```
dev.copy(pdf, 'SupplementalFigures/z.pdf', height=6,width=6);dev.off()
```

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

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

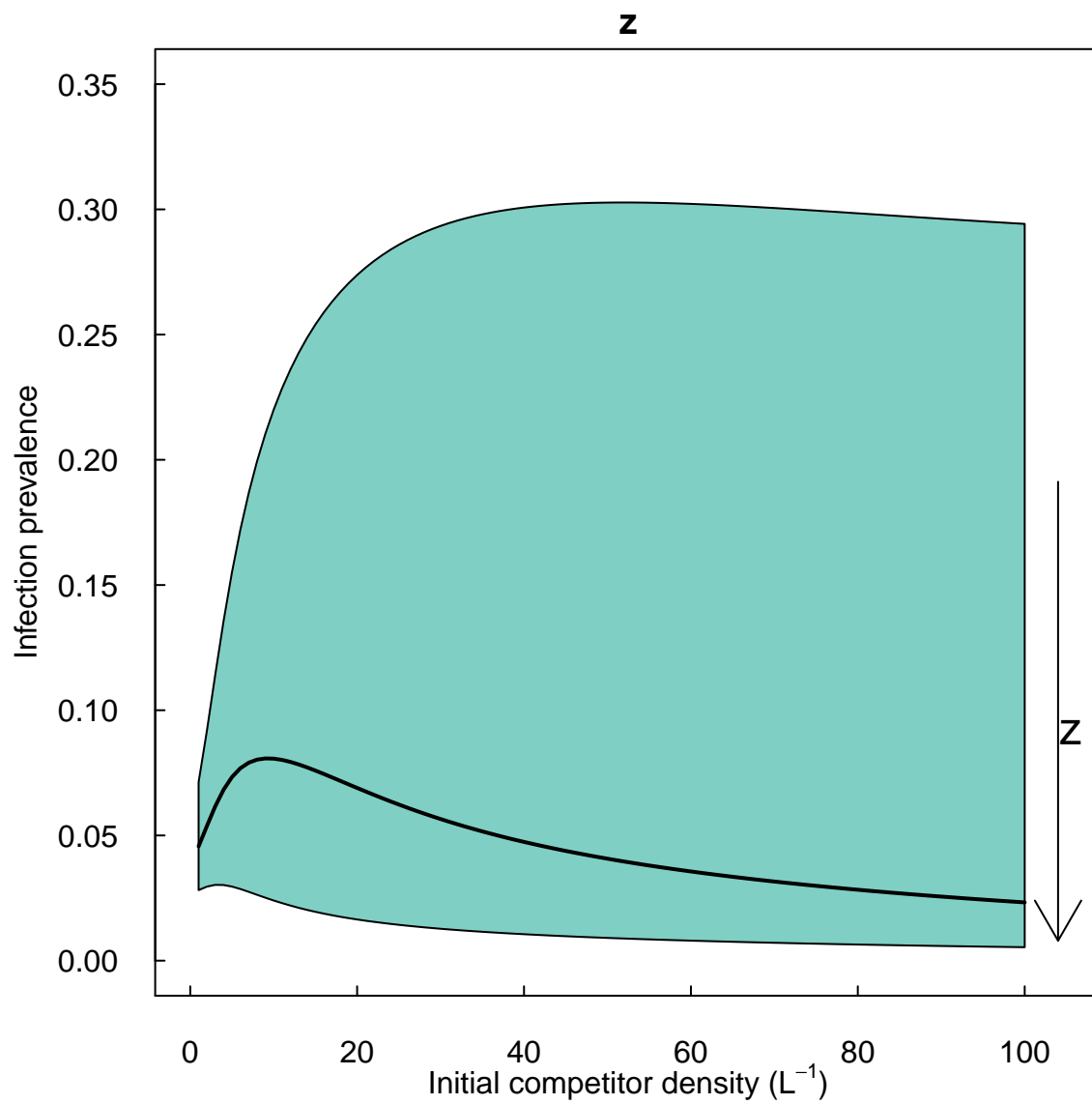


Figure 6:

Relative spore digestion (ζ_{CS})

```
zetaCSSensitive <- getSensitive(7, seq(0.3, 3, length.out=100), compN=100)

par(mar=c(3,4,1.5,0.5))
makePlot(zetaCSSensitive, col=cols[1], xlab='', ylab='Infection prevalence',
        main=expression(paste(zeta,'cs')), ylim=c(0,0.2), xlim=c(0,106.5))
mtext(expression(paste("Initial competitor density (L-1,"))), side=1, line=2)
lines(zetaCSSensitive[,21], lwd=2, col=1)
arrows(x0=103, y1 = min(zetaCSSensitive[nrow(zetaCSSensitive),]),
       y0=(0.75 * max(zetaCSSensitive[nrow(zetaCSSensitive),])))
text(106.5, (1.75 * mean(zetaCSSensitive[nrow(zetaCSSensitive),])),
     expression(paste(zeta,"cs")), cex=1.5)
```

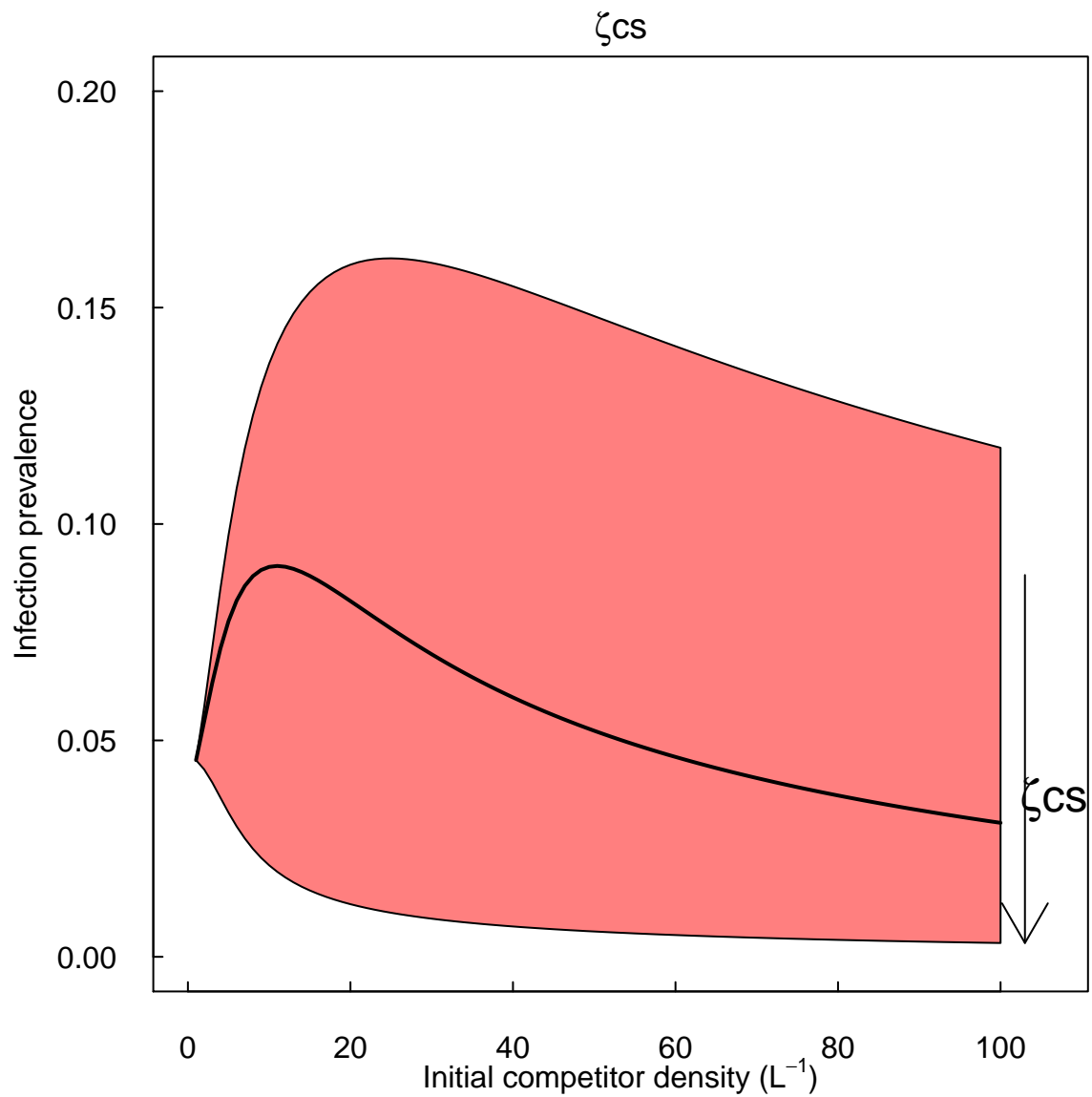


Figure 7:

```
dev.copy(pdf, 'SupplementalFigures/zetaCS.pdf', height=6,width=6);dev.off()
```

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

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

μIS

```
muISSensitive <- getSensitive(2, seq(1, 5, length.out=100), compN=100)

par(mar=c(3,4,1.5,0.5))
makePlot(muISSensitive, col=cols[1], xlab='', ylab='Infection prevalence',
         main=expression(paste(mu[IS])), ylim=c(0,0.1), xlim=c(0,110))

mtext(expression(paste("Initial competitor density (L"^{-1}),")), side=1, line=2)

lines(muISSensitive[,11], lwd=2, col=1)

#lines(muISSensitive[,27], lwd=2, col=1)
text(18.5, 0.086, expression(paste(mu[IS], ' = 2.05'))))

lines(muISSensitive[,1], col=1, lty=2)
text(106, (muISSensitive[100,1]), expression(paste(mu[IS], ' = 1 ')))

lines(muISSensitive[,76], col=1, lty=2)
text(106, (muISSensitive[100,76]), expression(paste(mu[IS], ' = 4 ')))
text(106, (.9*muISSensitive[100,100]), expression(paste(mu[IS], ' = 5 ')))

dev.copy(pdf, 'SupplementalFigures/muIS.pdf', height=6,width=6);dev.off()

## pdf
## 3

## pdf
## 2
```

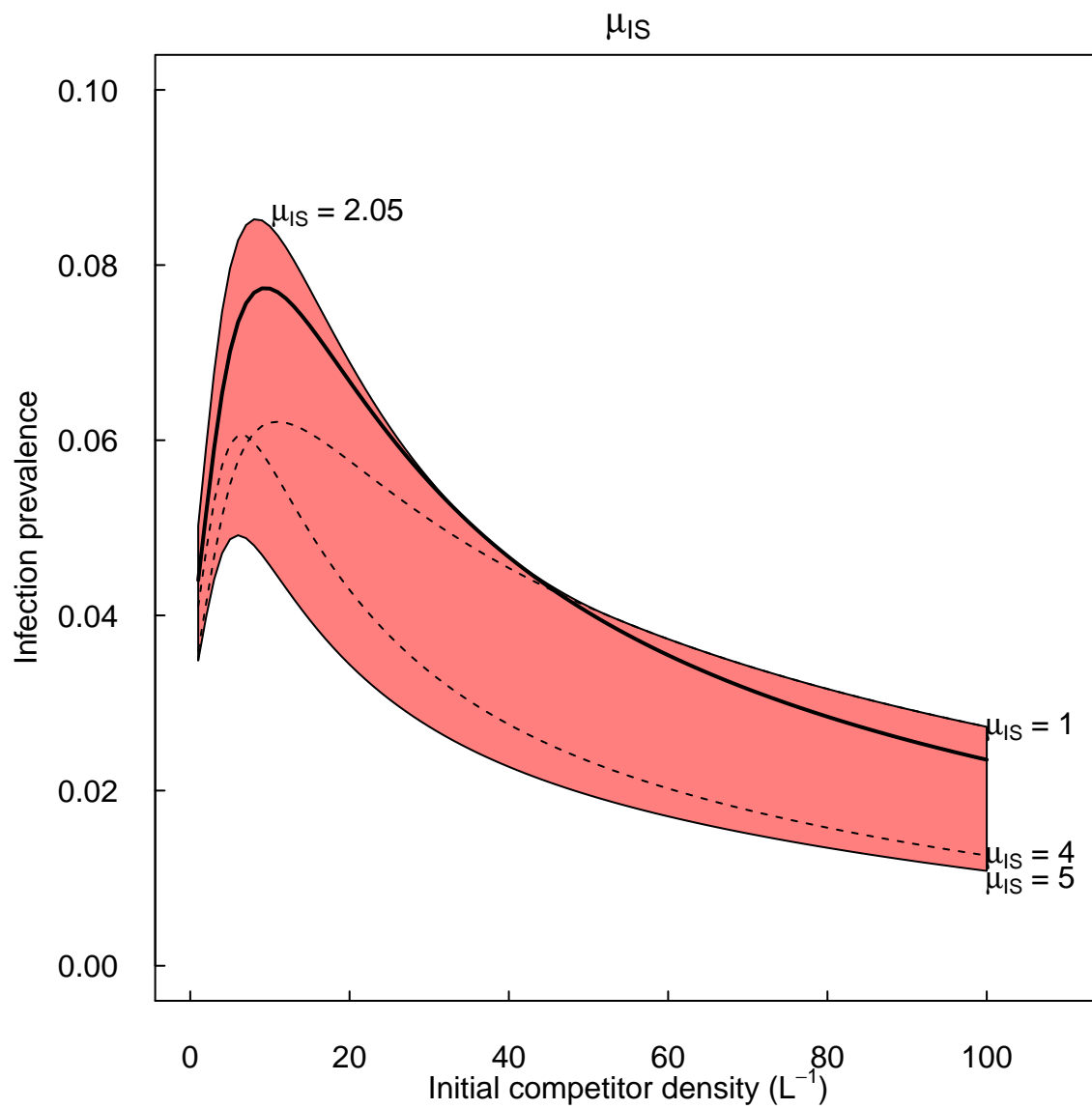


Figure 8:

γCS

```
gammaCSSensitive <- getSensitive(11, seq(0.7, 1.42, length.out=100), compN=100)

par(mar=c(3,4,1.5,0.5))
makePlot(gammaCSSensitive, col=cols[3], xlab='', ylab='Infection prevalence',
         main=expression(paste(gamma[CS])), ylim=c(0, 0.125), xlim=c(0,117))

mtext(expression(paste("Initial competitor density ( $L^{-1}$ ,"))), side=1, line=2)
lines(gammaCSSensitive[,98], lwd=2, col=1)

text(110, (0.95*gammaCSSensitive[100,100]), expression(paste(gamma[CS], ' = 1.4 ')))

lines(gammaCSSensitive[,8], col=1, lty=2)
text(110, (1.1*gammaCSSensitive[100,2]), expression(paste(gamma[CS], ' = 0.7 ')))

lines(gammaCSSensitive[,77], col=1, lty=2)
text(110, (1.05*gammaCSSensitive[100,77]), expression(paste(gamma[CS], ' = 1.25'))))

dev.copy(pdf, 'SupplementalFigures/gammaCS.pdf', height=6,width=6);dev.off()

## pdf
## 3

## pdf
## 2
```

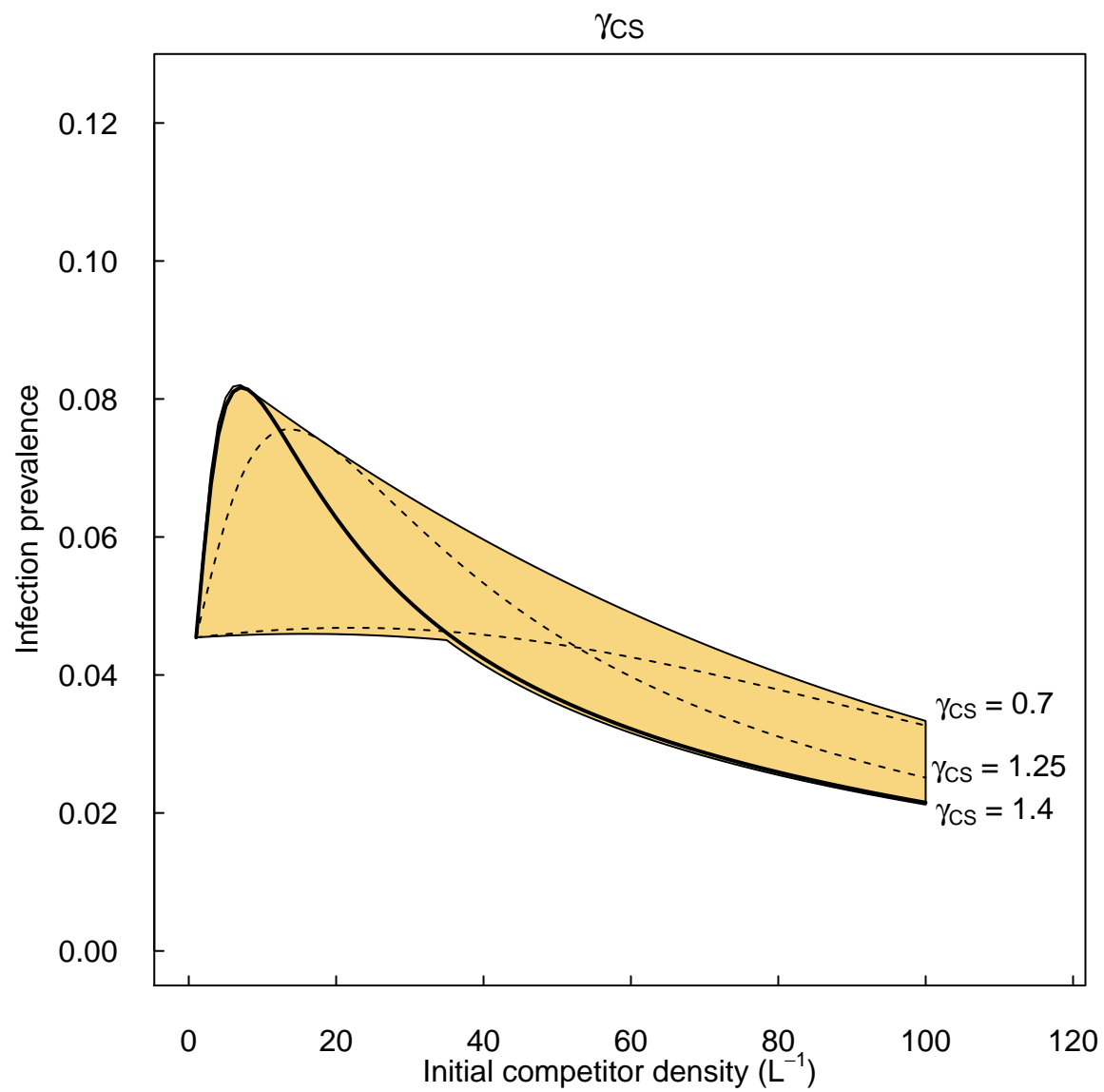


Figure 9:

w

```
wSensitive <- getSensitive(12, seq(2, 6, length.out=100), compN=100)

par(mar=c(3,4,1.5,0.5))
makePlot(wSensitive, col=cols[4], xlab='', ylab= 'Infection prevalence',
        main='w', ylim=c(0, 0.25), xlim=c(0,106.5))
mtext(expression(paste("Initial competitor density ( $L^{-1}$ )")), side=1, line=2)
lines(wSensitive[,50], lwd=2, col=1)
arrows(x0=103, y0 = min(wSensitive[nrow(wSensitive),]),
       y1=(0.75 * max(wSensitive[nrow(wSensitive),])))
text(106.5, (1.5 * mean(wSensitive[nrow(wSensitive),])), 'w', cex=1.5)
```

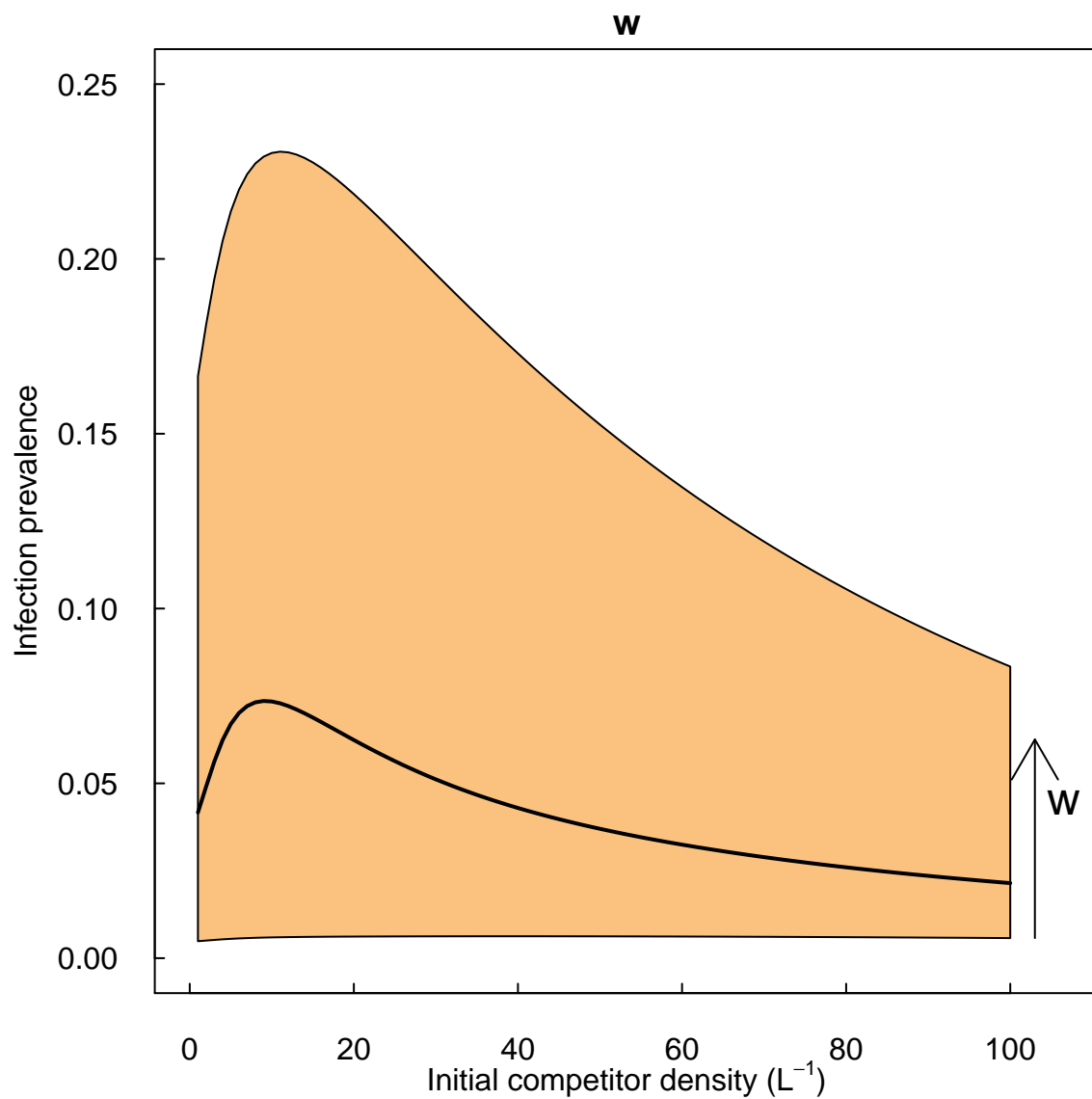


Figure 10:

```
dev.copy(pdf, 'SupplementalFigures/w.pdf', height=6,width=6);dev.off()
```

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

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