Analyzing Norovirus Data

Kayla Zhang, Damie Pak, Megan Greischar

2024-04-27

We use the R package metaDigitise to extract data points from figure 1C-E in Atmar *et al.*, 2008.

```
require(metaDigitise)

atmardata <- metaDigitise("Process Images", summary=FALSE)
atmardata <- rbind(atmardata$scatterplot$atmarplotc.jpg,
                   rbind(atmardata$scatterplot$atmarplotd.jpg,
                         atmardata$scatterplot$atmarplote.jpg))
```

We then fit a statistical model (Li & Handel, 2014, Holder & Beauchemin, 2011) that estimates viral load values over time to the shedding data we just extracted using the least squares method. There is a typo in Li & Handel, 2014–in the denominator, they have mistakenly written the term exp(p4 * (t-p3)) as exp(-p4 * (t-p3)). We make the assumption that viral load is a 1-to-1 mapping to viral shedding. We also subtract 19 hours each time we convert from days to hours to set time point = 0 as time of inoculation, since Atmar et al., 2008 defined study day 1 as beginning around 5-6 hours after inoculation.

```
generate.estimates <- function(t, pars) {

  p1 <- exp(pars[1])
  p2 <- exp(pars[2])
  p3 <- exp(pars[3])
  p4 <- exp(pars[4])

  estimate <- (2 * p1) / (exp(-p2*(t-p3)) + exp(p4*(t-p3)))

  return(log10(estimate))

}

sse.viralLoad <- function(pars, data) {
  vl <- log10(data$y)
  t <- data$x * 24 - 19

  estimates <- generate.estimates(t, pars)

  sse <- sum((estimates - vl)^2)

  if (sse > 1e20) {
    return(1e20)
  } else {
    return(sse)
  }

}
```

```
generate.possible.params <- function(data) {
  numguess <- 1e4

  fits <- data.frame(matrix(NA, nrow = numguess, ncol = 6))

  colnames(fits) <- c('p1','p2','p3','p4','convergence','sse')

  # These are ln(param) values
  for (g in 1:numguess) {
    p1 <- runif(1, -1, log(1e12))
    p2 <- runif(1, -1, 1)
    p3 <- runif(1, -1, log(4*24))
    p4 <- runif(1, -5, 1)

    params <- c(p1, p2, p3, p4)

    fit <- optim(params, sse.viralLoad, data=data)

    fits[g, ] <- c(fit$par, fit$convergence, fit$value)
  }

  return(fits)
}

get.params <- function(data) {

  fits <- generate.possible.params(data)

  minsse <- min(fits[fits$convergence==0, ]$sse)

  pars <- unlist((fits[fits$sse == minsse, ])[1:4])

  return (c(pars, minsse))
}
```

We define two endpoints for the fit: 1) stopping at the first local minimum in viral load after the peak, and 2) stopping at the first viral load value after the peak that is equal to or less than the inoculum dose/initial viral load.

```
make.param.df <- function(data, maxt) {

  participants <- unique(data$participant)

  participantparams <- matrix(nrow=length(participants), ncol=5)

  for (i in 1:length(participants)) {
    pars <- get.params(data[data$participant==participants[i],]
                       [1:(maxt[maxt$participant==participants[i],]$max.point),])

    participantparams[i,] <- c(participants[i], pars[1:4])
  }

  participantparams <- as.data.frame(participantparams)
  colnames(participantparams) <- c("participant", "p1", "p2",
```

```
                                "p3", "p4")

  return(participantparams)
}

atmardata <- read.csv("atmardata.csv")
participants <- unique(atmardata$participant)

# Method 1: Stopping at the first local minimum
maxt.firstmin <- data.frame(participant=participants,
                            max.point=rep(NA, length(participants)))

for (p in participants) {

  vldata <- atmardata[atmardata$participant==p,]

  t.peak <- which(vldata$y == max(vldata$y))

  points.after.peak <- vldata$y[-1:-t.peak]

  next.points <- append(vldata$y[-1:-(t.peak+1)], NA)

  local.min <- which(points.after.peak <= next.points)[1] + t.peak

  if (is.na(local.min)) {

    maxt.firstmin[maxt.firstmin$participant==p,]$max.point <- length(vldata$y)

  } else {

    maxt.firstmin[maxt.firstmin$participant==p,]$max.point <- local.min

  }
}

# Method 2: Stopping when viral load gets back down to the initial viral load
maxt.backtoinitial <- data.frame(participant=participants,
                                 max.point=rep(NA, length(participants)))

for (p in participants) {
  vldata <- atmardata[atmardata$participant==p,]

  t.peak <- which(vldata$y == max(vldata$y))

  points.after.peak <- vldata$y[-1:-t.peak]

  maxt.backtoinitial[maxt.backtoinitial$participant==p,]$max.point <-
    which(points.after.peak <= vldata$y[1])[1] + t.peak
}
```

Note that because this model is statistical, not deterministic, the parameters will fluctuate with each run.

```
params.localmin <- make.param.df(atmardata, maxt.firstmin)
```

```
params.backtoinitial <- make.param.df(atmardata, maxt.backtoinitial)
```

As a result, we saved the results of our first iteration in a csv file.

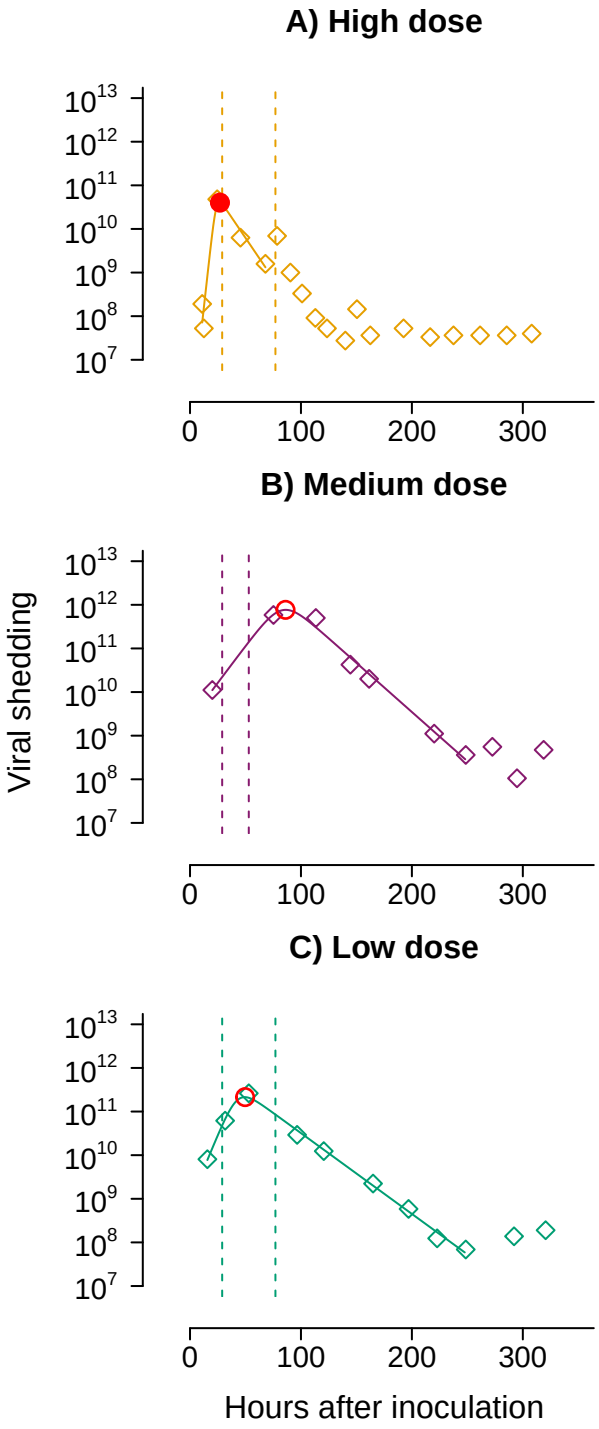
```
params.localmin <- read.csv("Atmar Parameters local minimum.csv")
params.backtoinitial <- read.csv("Atmar Parameters back to initial.csv")
```

Now we plot the trajectories of three participants (703, 722, 724) as an example.

```
plot.viralload <- function(data, pars, sympdata, max.modeltime, yhigh, cex,
                           xaxisLabs = T, yaxisLabs = T,
                           cex_labels, cex.xaxis, xlabel, ylabel, example) {


  if(unique(participantdata$dose) == '4800') {
    col <- "#E69F00"
    if (example) {title <- "A) High dose"}
  } else if (unique(participantdata$dose) == '48') {
    col <- "#871a6e"
    if (example) {title <- "B) Medium dose"}
  } else if (unique(participantdata$dose) == '4.8') {
    col <- "#009E73"
    if (example) {title <- "C) Low dose"}
  }

  t <- data$x * 24 - 19

  vl <- data$y

  # Plotting real VL
  plot(t, log10(vl), col=col, xlim=c(0, 350), ylim=c(7, yhigh),
     xlab="", ylab="", xaxt="n", yaxt="n", cex.lab=cex, cex.axis=cex,
     pch=5, font.lab=2, cex=cex, bty='n')

  # Plotting simulated VL
  linet <- seq(min(data$x)*24-19, data[max.modeltime,]$x*24-19, by=1)
  predictedvl <- generate.estimates(linet, pars)

  symp.onset <- sympdata$symp.onset * 24 - 19
  symp.end <- symp.onset + sympdata$symp.dur * 24

  lines(linet, predictedvl, col=col)

  abline(v = symp.onset, col=col, lty=2)
  abline(v = symp.end, col=col, lty=2)

  t.peak <- linet[which(predictedvl == max(predictedvl))]

  if (t.peak < symp.onset) {
    redpointstyle <- 19
    symp.timing <- "Pre-symptomatic"
  } else {
    redpointstyle <- 1
    symp.timing <- "Post-symptomatic"
  }
```

```
  if (!example) {
    title <- paste("ID:", unique(data$participant))
    text(x=300, y=13, labels=title, font=2)
    ylabel.line <- 4.5
    xlabel.line <- 3.5
  } else {
    mtext(title, font=2, line=2)
    ylabel.line <- 5.5
    xlabel.line <- 4
  }

  points(t.peak, max(predictedvl), pch=redpointstyle,
         col="red", lwd=cex, cex=cex+0.5)

  xLabs = NA
  if(xaxisLabs==T){
        xLabs = c(expression(0),
                  expression(100),
                  expression(200),
                  expression(300),
                  expression(400))}
  axis(1, cex.axis=cex.xaxis, at=seq(0, 400, by=100), labels = xLabs,
       line=1.25,
       mgp=c(2, 0.75, 0), tck=-0.04)

  yLabs = NA
  if(yaxisLabs==T){
      yLabs = c(expression(10^7),
                expression(10^8),
                expression(10^9),
                expression(10^10),
                expression(10^11),
                expression(10^12),
                expression(10^13),
                expression(10^14))}
  axis(2, cex.axis=cex, at=seq(7, 14, by=1),
       labels=yLabs,
       line=1.25,
       las=2,
       mgp=c(2, 1, 0), tck= -0.04)

  mtext(text=ylabel, side=2, line=ylabel.line, cex=cex_labels)

  mtext(text=xlabel, side=1, line=xlabel.line, cex=cex_labels)

}

atmarsymptoms <- read.csv('atmarsymptoms.csv')

participantexamples <- c('703', '722', '724')

par(mfrow=c(3, 1), mai=c(0.52, 0.82, 0.42, 0.1), oma=c(2, 3, 0, 0))
```

```r
for(p in participantexamples) {

  participantdata <- atmardata[atmardata$participant==p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant == p, ]
  params <- unlist(params.localmin[params.localmin$participant==p,][2:5])

  if(p==participantexamples[2]) {
    ylabel <- "Viral shedding"
    xlabel <- ""
  } else if (p == participantexamples[3]) {
    xlabel <- "Hours after inoculation"
    ylabel <- ""
  } else {
    xlabel <- ""
    ylabel <- ""
  }

  plot.viralload(data=participantdata,
                 pars=params,
                 sympdata=sympdata,
                 max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
                 yhigh=13,
                 cex=1.33,
                 cex_labels=1,
                 cex.xaxis=1.33,
                 xlabel=xlabel,
                 ylabel=ylabel,
                 example=TRUE)
}
```

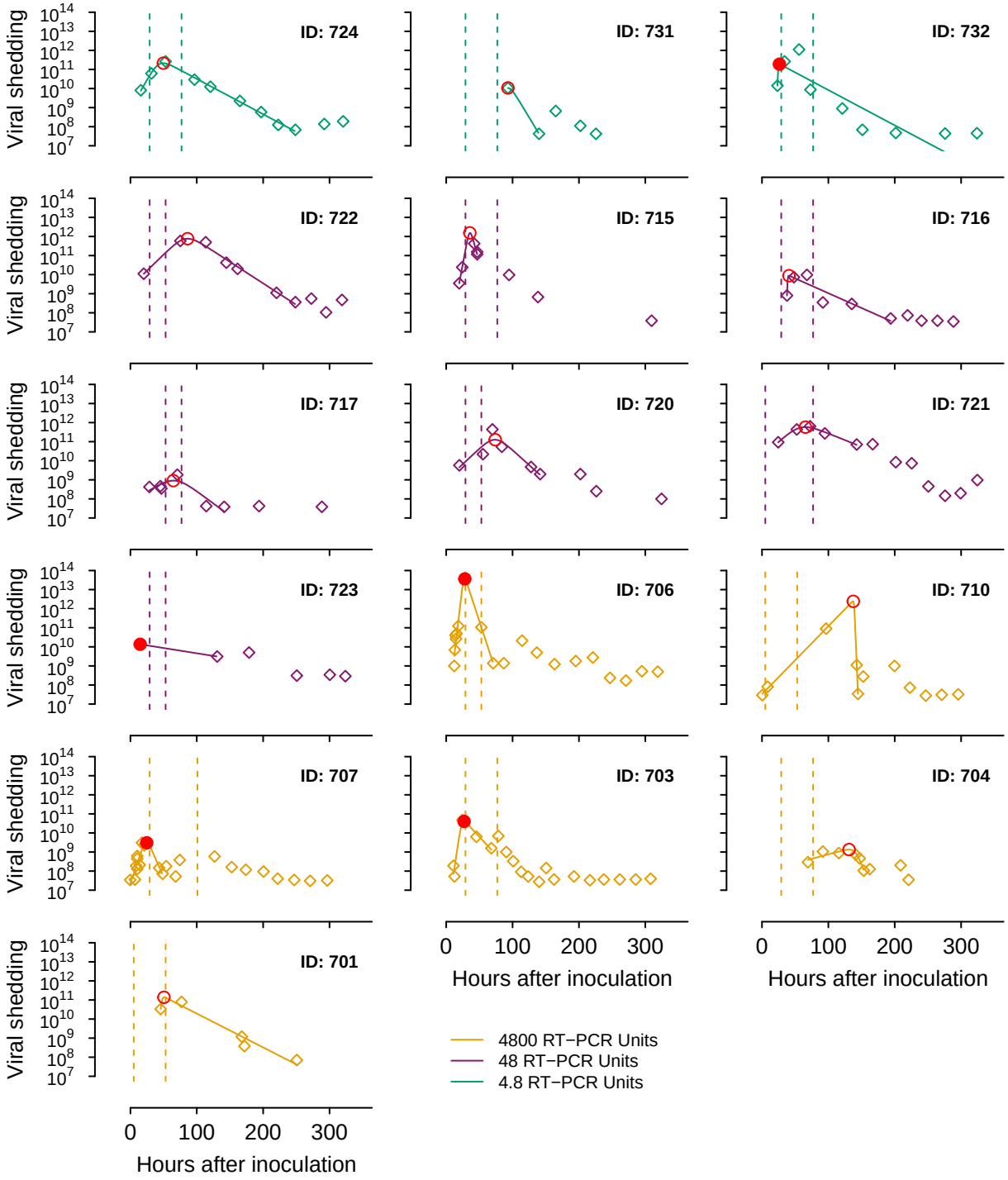


We then plot the trajectories of each participant using both sets of endpoints. Since they produce qualitatively similar fits, we arbitrarily use the first local minimum in pathogen load as the endpoint going forward.

```
participants.by.dose <- unique(atmardata$participant[order(atmardata$dose)])

par(mfrow=c(6, 3), mai=c(0.22, 0.42, 0.22, 0.22), oma=c(3.5, 4, 0, 0),
    mar=c(1, 2.1, 1, 1))
```

```
for(i in 1:length(participants.by.dose)) {

  p <- participants.by.dose[i]
  participantdata <- atmardata[atmardata$participant==p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant==p, ]
  params <- unlist(params.localmin[params.localmin$participant==p,][2:5])
  dose <- unique(participantdata$dose)

  if (i %in% c(1, 4, 7, 10, 13)) {
    plot.viralload(
      data=participantdata,
      pars=params,
      sympdata=sympdata,
      max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
      yhigh=14,
      cex=1,
      xaxisLabs=F,
      cex_labels=0.8,
      cex.xaxis=1.25,
      xlabel="",
      ylabel="Viral shedding",
      example=FALSE)
  } else if (i %in% c(14, 15)) {
     plot.viralload(
      data=participantdata,
      pars=params,
      sympdata=sympdata,
      max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
      yhigh=14,
      cex=1,
      yaxisLabs=F,
      cex_labels=0.8,
      cex.xaxis=1.25,
      xlabel="Hours after inoculation",
      ylabel="",
      example=FALSE)
  } else if (i==16) {
    plot.viralload(
      data=participantdata,
      pars=params,
      sympdata=sympdata,
      max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
      yhigh=14,
      cex=1,
      cex_labels=0.8,
      cex.xaxis=1.25,
      xlabel="Hours after inoculation",
      ylabel="Viral shedding",
      example=FALSE)
  } else {
    plot.viralload(
    data=participantdata,
    pars=params,
```

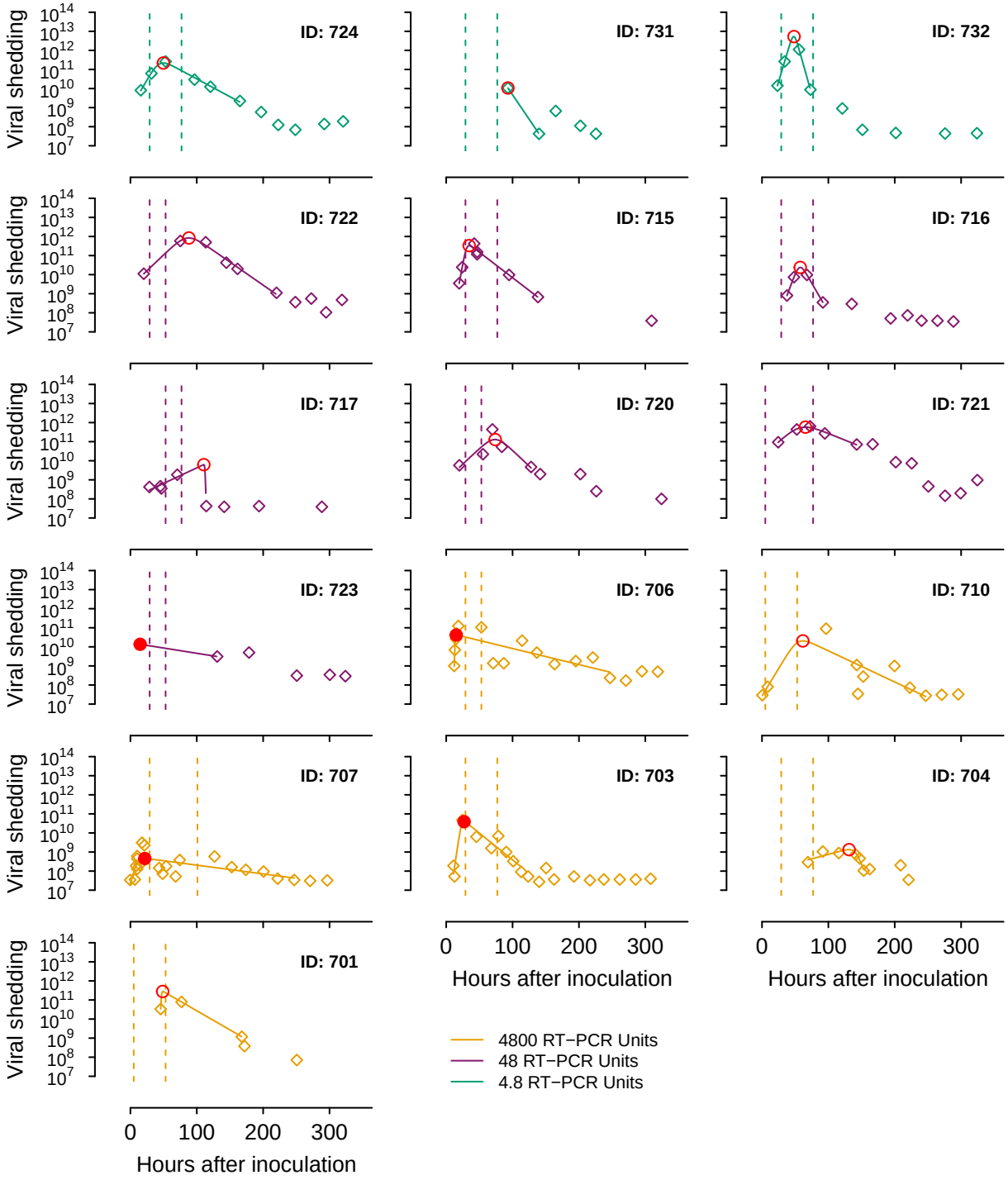
```
    sympdata=sympdata,
    max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
    yhigh=14,
    cex=1,
    xaxisLabs=F,
    yaxisLabs=F,
    cex_labels=0.8,
    cex.xaxis=1.25,
    xlabel="",
    ylabel="",
    example=FALSE)
  }
}

par(mar=c(0, 0, 0, 0))
plot.new()
legend(x = "bottom",
       legend = c("4800 RT-PCR Units",
                  "48 RT-PCR Units",
                  "4.8 RT-PCR Units"),
       col = c("#E69F00",
               "#871a6e",
               "#009E73"),
       lty = 1, bty='n')
plot.new()
```




```
par(mfrow=c(6, 3), mai=c(0.22, 0.42, 0.22, 0.22), oma=c(3.5, 4, 0, 0),
    mar=c(1, 2.1, 1, 1))

for(i in 1:length(participants.by.dose)) {

  p <- participants.by.dose[i]
  participantdata <- atmardata[atmardata$participant==p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant==p, ]
```

```
params <- unlist(params.backtoinitial[params.backtoinitial$participant==p,][2:5])
dose <- unique(participantdata$dose)

if (i %in% c(1, 4, 7, 10, 13)) {
  plot.viralload(
    data=participantdata,
    pars=params,
    sympdata=sympdata,
    max.modeltime=maxt.backtoinitial[maxt.backtoinitial$participant==p,]$max.point,
    yhigh=14,
    cex=1,
    xaxisLabs=F,
    cex_labels=0.8,
    cex.xaxis=1.25,
    xlabel="",
    ylabel="Viral shedding",
    example=FALSE)
} else if (i %in% c(14, 15)) {
   plot.viralload(
    data=participantdata,
    pars=params,
    sympdata=sympdata,
    max.modeltime=maxt.backtoinitial[maxt.backtoinitial$participant==p,]$max.point,
    yhigh=14,
    cex=1,
    yaxisLabs=F,
    cex_labels=0.8,
    cex.xaxis=1.25,
    xlabel="Hours after inoculation",
    ylabel="",
    example=FALSE)
} else if (i==16) {
  plot.viralload(
    data=participantdata,
    pars=params,
    sympdata=sympdata,
    max.modeltime=maxt.backtoinitial[maxt.backtoinitial$participant==p,]$max.point,
    yhigh=14,
    cex=1,
    cex_labels=0.8,
    cex.xaxis=1.25,
    xlabel="Hours after inoculation",
    ylabel="Viral shedding",
    example=FALSE)
} else {
  plot.viralload(
  data=participantdata,
  pars=params,
  sympdata=sympdata,
  max.modeltime=maxt.backtoinitial[maxt.backtoinitial$participant==p,]$max.point,
  yhigh=14,
  cex=1,
  xaxisLabs=F,
```

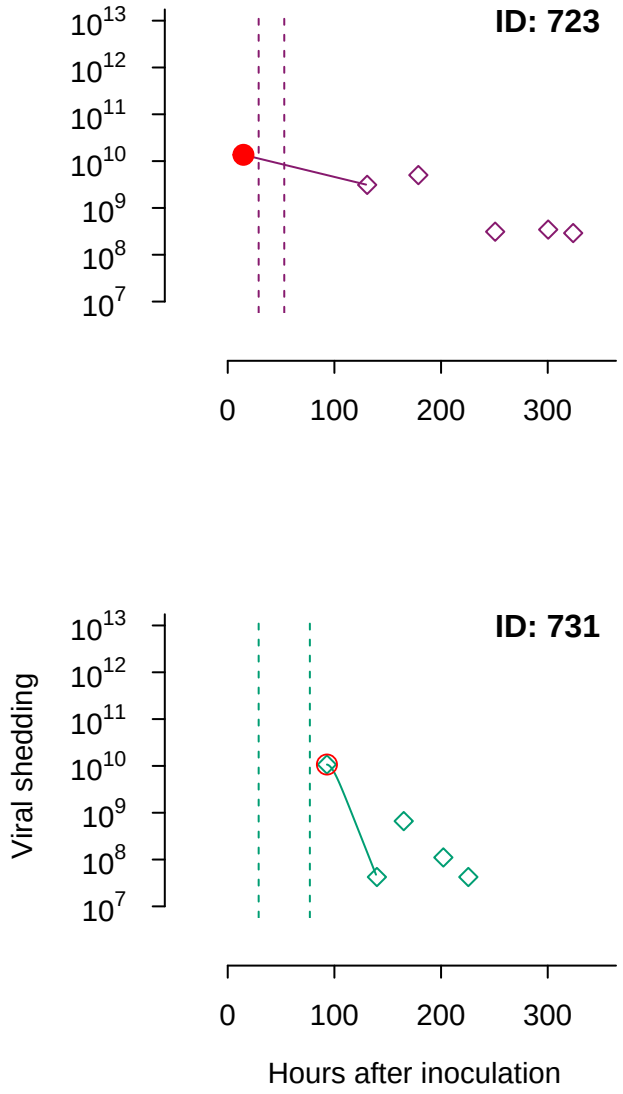
```
    yaxisLabs=F,
    cex_labels=0.8,
    cex.xaxis=1.25,
    xlabel="",
    ylabel="",
    example=FALSE)
  }
}

par(mar=c(0, 0, 0, 0))
plot.new()
legend(x = "bottom",
       legend = c("4800 RT-PCR Units",
                  "48 RT-PCR Units",
                  "4.8 RT-PCR Units"),
       col = c("#E69F00",
               "#871a6e",
               "#009E73"),
       lty = 1, bty='n')
plot.new()
```



Because participants 723 and 731 do not have clear peaks in pathogen load, we exclude them from any future calculations using our conservative measure of pre-symptomatic transmission, which is that peak pathogen load is reached before symptom onset.

```
excluded.participants <- c('723', '731')

par(mfrow=c(2, 1), mai=c(0.75, 0.75, 0.82, 0.25), oma=c(1, 2, 0, 0))

for(i in 1:length(excluded.participants)) {
```

```
  p <- excluded.participants[i]
  participantdata <- atmardata[atmardata$participant==p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant == p, ]
  params <- unlist(params.localmin[params.localmin$participant==p,][2:5])

  if(i == 2) {
    ylabel <- "Viral shedding"
    xlabel <- "Hours after inoculation"
  } else {
    xlabel <- ""
    ylabel <- ""
  }

  plot.viralload(data=participantdata,
                 pars=params,
                 sympdata=sympdata,
                 max.modeltime=maxt.firstmin[maxt.firstmin$participant==p,]$max.point,
                 yhigh=13,
                 cex=0.9,
                 cex_labels=0.9,
                 cex.xaxis=0.9,
                 xlabel=xlabel,
                 ylabel=ylabel,
                 example=FALSE)
  if(i==2) {
    legend(x = 100, y = 10^12,
       legend = c("48 RT-PCR Units",
                  "4.8 RT-PCR Units"),
       col = c("#871a6e",
               "#009E73"),
       lty = 1, bty='n', cex=0.8)
  }
}
```



Here we calculate the delay between symptom onset and peak pathogen load for each participant in the Atmar *et al.*, 2008 study.

```
participants.conservative <- participants[participants != 723 &
                                          participants != 731]

delays.conservative <- data.frame("participant"=participants.conservative,
                    "delay"=rep(0, length(participants.conservative)),
                    "dose"=rep(0, length(participants.conservative)))

for (p in participants.conservative) {

  vldata <- atmardata[atmardata$participant == p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant == p, ]
```

```
  dose <- unique(vldata$dose)

  params <- unlist(params.localmin[params.localmin$participant==p, 2:5])

  t <- seq(min(vldata$x)*24-19,
        vldata[maxt.firstmin[maxt.firstmin$participant==p,]$max.point,]$x*24-19,
        by=1)

  predictedvl <- generate.estimates(t, params)

  peak.vl.time <- t[which(predictedvl==max(predictedvl))]

  participantdelay <- peak.vl.time - (sympdata$symp.onset * 24 - 19)

  delays.conservative[delays.conservative$participant==p, ]$delay <- participantdelay

  delays.conservative[delays.conservative$participant==p, ]$dose <- dose

}
```

We run a linear regression between the delay between symptom onset and peak pathogen load and each of the 4 parameters from the statistical model as well as between viral growth and decay rate. Only p2 and p4, which represent pathogen replication rate and time of peak pathogen load, respectively, were found to have significant correlations with delay. They are also significantly correlated to each other.

```
params.and.delays.conservative <- merge(params.localmin, delays.conservative)

lm.fit.p1.delay.conservative <-
  lm(delay/24 ~ p1, data=params.and.delays.conservative)
p1.delay.conservative.pval <-
  summary(lm.fit.p1.delay.conservative)$coefficients[2,4]
p1.delay.conservative.pval
```

```
## [1] 0.8395396
```

```
lm.fit.p2.delay.conservative <-
  lm(delay/24 ~ p2, data=params.and.delays.conservative)
p2.delay.conservative.pval <-
  summary(lm.fit.p2.delay.conservative)$coefficients[2,4]
p2.delay.conservative.pval
```

```
## [1] 0.01819126
```

```
lm.fit.p3.delay.conservative <-
  lm(delay/24 ~ p3, data=params.and.delays.conservative)
p3.delay.conservative.pval <-
  summary(lm.fit.p3.delay.conservative)$coefficients[2,4]
p3.delay.conservative.pval
```

```
## [1] 6.208749e-06
```

```
lm.fit.p4.delay.conservative <-
  lm(delay/24 ~ p4, data=params.and.delays.conservative)
p4.delay.conservative.pval <-
  summary(lm.fit.p4.delay.conservative)$coefficients[2,4]
p4.delay.conservative.pval
```

```
## [1] 0.10205
```

```
# Viral growth vs. decay rate
lm.fit.p4.p2.conservative <-
  lm(p4 ~ p2, data=params.and.delays.conservative)
p4.p2.conservative.pval <-
  summary(lm.fit.p4.p2.conservative)$coefficients[2,4]
p4.p2.conservative.pval
```

```
## [1] 0.3998268
```

```
# Test for collinearity
lm.fit.p2.p3.conservative <-
  lm(p2 ~ p3, data=params.and.delays.conservative)
p2.p3.conservative.pval <-
  summary(lm.fit.p2.p3.conservative)$coefficients[2,4]
p2.p3.conservative.pval
```

```
## [1] 0.001477611
```

Using a non-conservative measure of pre-symptomatic transmission, ie shedding onset values as reported in the study, we calculate the delay between shedding onset and symptom onset (the length of the latency period). For simplicity's sake, a negative delay still indicates pre-symptomatic transmission.

```
atmarshedding <- read.csv("atmarshedding.csv")

delays.nonconservative <- data.frame("participant"=participants,
                     "delay"=rep(0, length(participants)),
                     "dose"=rep(0, length(participants)))

for (p in participants) {

  sheddata <- atmarshedding[atmarshedding$participant == p, ]
  sympdata <- atmarsymptoms[atmarsymptoms$participant == p, ]
  dose <- unique(sheddata$dose)

  participantdelay <- sheddata$shed.onset - sympdata$symp.onset

  delays.nonconservative[delays.nonconservative$participant==p, ]$delay <- participantdelay

  delays.nonconservative[delays.nonconservative$participant==p, ]$dose <- dose

}
```

As before, we test for any significant correlations between the statistical model parameters and the duration of the latency period as well as between viral growth and decay rate. Just as with the conservative measure of pre-symptomatic transmission, there is a significant correlation between both viral replication rate and time of peak pathogen load with the duration of the latency period. We test for collinearity with another linear regression, which returns positive.

```
params.and.delays.nonconservative <- merge(params.localmin, delays.nonconservative)

lm.fit.p1.delay.nonconservative <-
  lm(delay ~ p1, data=params.and.delays.nonconservative)
p1.delay.nonconservative.pval <-
  summary(lm.fit.p1.delay.nonconservative)$coefficients[2,4]
p1.delay.nonconservative.pval
```

```
## [1] 0.5842455
```

```
lm.fit.p2.delay.nonconservative <-
  lm(delay ~ p2, data=params.and.delays.nonconservative)
p2.delay.nonconservative.pval <-
  summary(lm.fit.p2.delay.nonconservative)$coefficients[2,4]
p2.delay.nonconservative.pval
```

```
## [1] 0.006357595
```

```
lm.fit.p3.delay.nonconservative <-
  lm(delay ~ p3, data=params.and.delays.nonconservative)
p3.delay.nonconservative.pval <-
  summary(lm.fit.p3.delay.nonconservative)$coefficients[2,4]
p3.delay.nonconservative.pval
```

```
## [1] 0.002999094
```

```
lm.fit.p4.delay.nonconservative <-
  lm(delay ~ p4, data=params.and.delays.nonconservative)
p4.delay.nonconservative.pval <-
  summary(lm.fit.p4.delay.nonconservative)$coefficients[2,4]
p4.delay.nonconservative.pval
```

```
## [1] 0.309953
```

```
# Viral growth vs. decay rate
lm.fit.p4.p2.nonconservative <-
  lm(p4 ~ p2, data=params.and.delays.nonconservative)
p4.p2.nonconservative.pval <-
  summary(lm.fit.p4.p2.nonconservative)$coefficients[2,4]
p4.p2.nonconservative.pval
```

```
## [1] 0.1374857
```

```
# Test for collinearity
lm.fit.p2.p3.nonconservative <-
  lm(p2 ~ p3, data=params.and.delays.nonconservative)
p2.p3.nonconservative.pval <-
  summary(lm.fit.p2.p3.nonconservative)$coefficients[2,4]
p2.p3.nonconservative.pval
```

```
## [1] 0.0003224078
```

We also run a linear regression between pre-symptomatic transmission potential, measured as average shedding prior to symptoms, and time of symptom onset. There is no correlation.

```
transmission.latent <- data.frame("participant"=participants.conservative,
                                  "dose"=rep(0, length(participants.conservative)),
                    "transmission_potential"=rep(0, length(participants.conservative)),
                    "symptom_onset"=rep(0, length(participants.conservative)))

for (i in 1:length(participants.conservative)) {
  p <- participants.conservative[i]

  transmission.latent[i,]$participant <- p

  symp.onset <- atmarsymptoms[atmarsymptoms$participant == p,]$symp.onset
```

```
  transmission.latent[i,]$symptom_onset <- symp.onset

  params <- unlist(params.localmin[params.localmin$participant == p,][2:5])

  vldata <- atmardata[atmardata$participant == p,]

  transmission.latent[i,]$dose <- unique(vldata$dose)

  shed.onset <- min(vldata$x) * 24 - 19

  if(shed.onset > symp.onset * 24 - 19) {
    avg.shedding.latent <- 0
  } else {
    t <- seq(min(vldata$x)*24-19, symp.onset*24-19, by=1)
    vl.latent <- 10^generate.estimates(t, params)
    avg.shedding.latent <- log10(mean(vl.latent))
  }

  transmission.latent[i,]$transmission_potential <- avg.shedding.latent
}

transmission.latent.fit <- lm(transmission_potential ~ symptom_onset,
                              data = transmission.latent)
transmission.latent.pval <-
  summary(transmission.latent.fit)$coefficients[2,4]

transmission.latent.pval
```

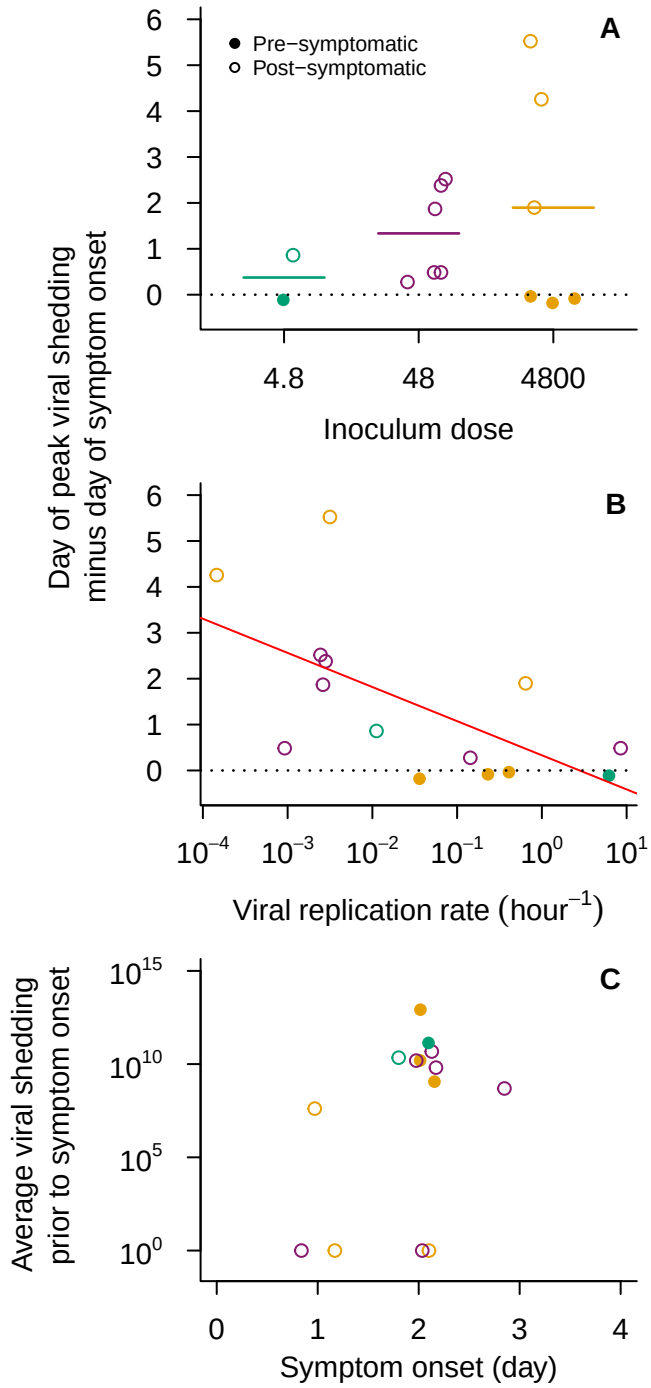
```
## [1] 0.1005292
```

We plot delay vs. inoculum dose, delay vs. viral growth rate, and pre-symptomatic transmission potential vs. symptom onset in a 3-panel figure.

```
id <- delays.conservative$dose
id[id==4.8] <- 1
id[id==48] <- 2
id[id==4800] <- 3

participant.params <- params.localmin[params.localmin$participant != 723 &
                                        params.localmin$participant != 731,]
transmission.latent.delays.data <- merge(transmission.latent, delays.conservative)

colors <- c("#009E73", "#871a6e","#E69F00")
cex <- 1.33
cex_labels <- .95
panellabelratio <- .975

{
  par(mfrow=c(3, 1), oma=c(0.5, 4, 0, 0), mai=c(0.6, 0.6, 0.2, 0.1))

  # Panel A: Delay between symptom onset and peak viral shedding vs. inoculum
  # dose

  plot(jitter(id), delays.conservative$delay/24,
          col=colors[factor(id)], xlim=c(.5, 3.5),
```

```
        ylim=c(-0.5, 6), cex.axis=cex, cex=cex,
        pch=ifelse(delays.conservative$delay < 0, 16, 1),
        xaxt="n",
        ylab = "",
        xlab = "",
     mgp=c(0, 1.5, 0), tck=-0.04, las=2, bty='l')
axis(1, at=seq(1,3), labels=c("4.8", "48", "4800"), cex.axis=cex,
     mgp=c(0, 1.5, 0), tck=-0.04)
mtext(text="Day of peak viral shedding \n minus day of symptom onset",
      side=2, line=-1, cex=cex_labels, outer=TRUE, adj=0.75)
mtext(text="Inoculum dose", side=1, line=3.5, cex=cex_labels)

#Average delays per infectious dose
segments(0.7,
         mean(delays.conservative[delays.conservative$dose == 4.8,]$delay/24),
         1.3, col=colors[1], lwd=cex)
segments(1.7,
         mean(delays.conservative[delays.conservative$dose == 48,]$delay/24),
         2.3, col=colors[2], lwd=cex)
segments(2.7,
         mean(delays.conservative[delays.conservative$dose == 4800,]$delay/24),
         3.3, col=colors[3], lwd=cex)

#Pre-symptomatic vs post-symptomatic line
abline(h=0, lty=3, lwd=cex)

text(panellabelratio * 3 + .5, panellabelratio * 7 - 1, labels="A", cex=cex, font=2)

 legend(0.5, 6, legend=c("Pre-symptomatic", "Post-symptomatic"),
       pch=c(19, 1, NA), col=c("black", "black"), cex=cex_labels, bty="n")

# Panel B: Delay between symptom onset and peak viral shedding vs. viral
# growth rate

plot(participant.params[order(participant.params$participant),]$p2,
   delays.conservative[order(delays.conservative$participant),]$delay/24,
   col=colors[factor(delays.conservative[order(delays.conservative$participant),]$dose)],
   pch=ifelse(delays.conservative[order(delays.conservative$participant),]$delay < 0, 16, 1),
   cex=cex, cex.axis=cex,
   ylab="", xlab="", xaxt="n",
   ylim=c(-0.5, 6),
   mgp=c(0, 1.5, 0), tck=-0.04, las=2, bty='l')

axis(1, at=seq(-4, 1), labels=c(expression(10^-4),
                                expression(10^-3),
                                expression(10^-2),
                                expression(10^-1),
                                expression(10^0),
                                expression(10^1)), cex.axis=cex,
     mgp=c(0, 1.5, 0), tck=-0.04)

abline(lm.fit.p2.delay.conservative, col="red")
```

```
  abline(h=0, lty=3, lwd=cex)

  mtext(text=expression(Viral~replication~rate~(hour^-1)),
        side=1, line=4, cex=cex_labels)

  text(panellabelratio * 5 - 4, panellabelratio * 7 - 1, labels="B", cex=cex, font=2)

  # Panel C: Pre-symptomatic transmission potential vs. symptom onset

  plot(transmission_potential ~ jitter(symptom_onset), data=transmission.latent.delays.data,
          col=colors[factor(dose)], xlim=c(0, 4),
        pch=ifelse(delay < 0, 16, 1),
          ylim=c(-1, 15), cex.axis=cex, cex=cex,
          ylab = "",
          xlab = "",
       xaxt="n",
       yaxt="n", bty='l')

  axis(1, cex.axis=cex, mgp=c(0, 1.5, 0), tck=-0.04)
  axis(2, cex.axis=cex, mgp=c(0, 1.5, 0), tck=-0.04, at=seq(0, 15, by=5),
       labels=c(expression(10^0),
                expression(10^5),
                expression(10^10),
                expression(10^15)),
       las=2)

  mtext(text="Average viral shedding \n prior to symptom onset",
        side=2, line=5, cex=cex_labels)
  mtext(text="Symptom onset (day)", side=1, line=3, cex=cex_labels)

  text(panellabelratio * 4, panellabelratio * 16 - 1, labels="C", cex=cex, font=2)
}
```

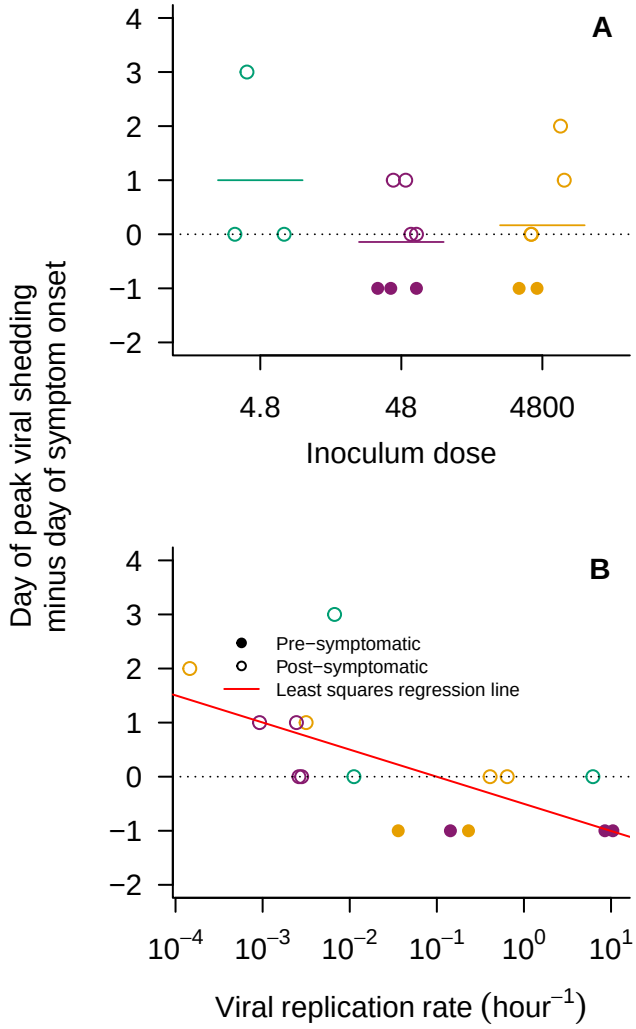


We again plot delay vs. inoculum dose and delay vs. viral growth rate using the non-conservative measure of pre-symptomatic transmission.

```
id.nonconservative <- delays.nonconservative$dose
id.nonconservative[id.nonconservative==4.8] <- 1
id.nonconservative[id.nonconservative==48] <- 2
id.nonconservative[id.nonconservative==4800] <- 3
```

```
cex <- 0.9
cex_labels <- 0.9
panellabelratio <- .975

{
  par(mfrow=c(2, 1), oma=c(1.75, 1, 0, 0), mai=c(0.5, 0.82, 0.5, 0.1))

  # Panel A: Delay between symptom onset and peak viral shedding vs. inoculum
  # dose

    plot(jitter(id.nonconservative), delays.nonconservative$delay,
          col=colors[factor(id.nonconservative)], xlim=c(.5, 3.5),
          ylim=c(-2, 4), cex.axis=cex, cex=cex,
          pch=ifelse(delays.nonconservative$delay < 0, 16, 1),
          xaxt="n",
          ylab = "",
          xlab = "",
       mgp=c(0, 0.8, 0), tck=-0.04, las=2, bty='l')

  axis(1, at=seq(1,3), labels=c("4.8", "48", "4800"), cex.axis=cex,
       mgp=c(0, 0.8, 0), tck=-0.04)
  mtext(text="Day of peak viral shedding \n minus day of symptom onset",
        side=2, line=-1.5, cex=cex_labels, outer=TRUE)
  mtext(text="Inoculum dose", side=1, line=2, cex=cex_labels)

    #Average delays per inoculum dose
  segments(0.7,
           mean(delays.nonconservative[delays.nonconservative$dose == 4.8,]$delay),
           1.3, col=colors[1], lwd=cex)
  segments(1.7,
           mean(delays.nonconservative[delays.nonconservative$dose == 48,]$delay),
           2.3, col=colors[2], lwd=cex)
  segments(2.7,
           mean(delays.nonconservative[delays.nonconservative$dose == 4800,]$delay),
           3.3, col=colors[3], lwd=cex)

  #Pre-symptomatic vs post-symptomatic line
  abline(h=0, lty=3, lwd=cex)

  text(panellabelratio * 3 + .5, panellabelratio * 6 - 2, labels="A", cex=cex, font=2)

  # Panel B: Delay between symptom onset and peak viral shedding vs. viral
  # growth rate

  plot(params.localmin[order(params.localmin$participant),]$p2,
     delays.nonconservative[order(delays.nonconservative$participant),]$delay,
     col=colors[factor(delays.nonconservative[order(delays.nonconservative$participant),]$dose)],
     pch=ifelse(delays.nonconservative[order(delays.nonconservative$participant),]$delay < 0, 16, 1),
     cex=cex, cex.axis=cex,
     ylab="", xlab="", xaxt="n",
     ylim=c(-2, 4),
     mgp=c(0, 0.8, 0), tck=-0.04, las=2, bty='l')
```

```
  axis(1, at=seq(-4, 1), labels=c(expression(10^-4),
                                  expression(10^-3),
                                  expression(10^-2),
                                  expression(10^-1),
                                  expression(10^0),
                                  expression(10^1)), cex.axis=cex,
       mgp=c(0, 0.8, 0), tck=-0.04)

  abline(lm.fit.p2.delay.nonconservative, col="red")

  abline(h=0, lty=3, lwd=cex)

  mtext(text=expression(Viral~replication~rate~(hour^-1)),
        side=1, line=2.5, cex=cex_labels)

    text(panellabelratio * 5 - 4, panellabelratio * 6 - 2, labels="B", cex=cex, font=2)

    legend(-3.5, 2.9, legend=c("Pre-symptomatic", "Post-symptomatic",
                           "Least squares regression line"), pch=c(19, 1, NA),
                           lty=c(NA, NA, 1), col=c("black", "black", "red"),
       cex=0.6, bty="n")
}
```



Lastly, we break down the correlation between pre-symptomatic transmission and viral growth rate by examining the relationships between time of peak shedding and viral growth rate as well as between symptom onset and viral growth rate.

```
participant.peaktimes <- data.frame("participant"=participants.conservative,
                    "peaktime"=rep(0, length(participants.conservative)),
                    "dose"=rep(0, length(participants.conservative)))

for (p in participants.conservative) {

  vldata <- atmardata[atmardata$participant == p, ]

  t <- seq(min(vldata$x)*24-19,
        vldata[maxt.firstmin[maxt.firstmin$participant==p,]$max.point,]$x*24-19,
        by=1)

  params <- unlist(params.localmin[params.localmin$participant==p, 2:5])
```

```
  predictedvl <- generate.estimates(t, params)

  participant.peaktime <- t[which(predictedvl==max(predictedvl))]

  dose <- unique(vldata$dose)

  participant.peaktimes[participant.peaktimes$participant==p, ]$dose <- dose

  participant.peaktimes[participant.peaktimes$participant==p, ]$peaktime <-
    participant.peaktime
}

params.peaktime.data <- merge(merge(participant.params, participant.peaktimes),
                              delays.conservative)
lm.fit.p2.peaktime <- lm(peaktime/24 ~ p2, data=params.peaktime.data)
p2.peaktime.pval <- summary(lm.fit.p2.peaktime)$coefficients[2,4]
p2.peaktime.pval
```

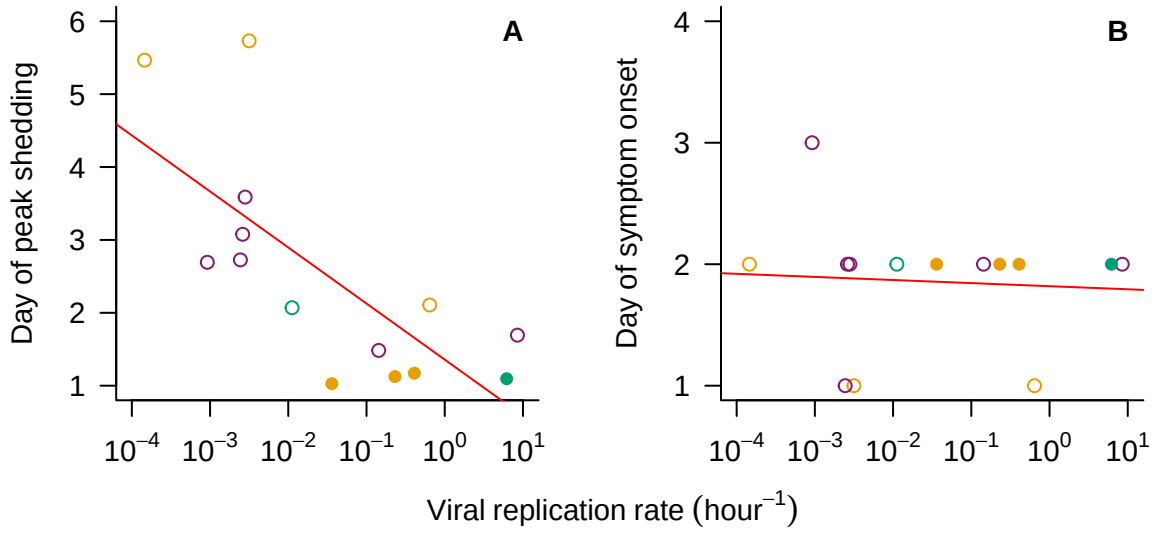
```
## [1] 0.002885796
```

```
params.symp.data <- merge(merge(participant.params, atmarsymptoms),
                          delays.conservative)
lm.fit.p2.symponset <- lm(symp.onset ~ p2, data=params.symp.data)
p2.symponset.pval <- summary(lm.fit.p2.symponset)$coefficients[2,4]
p2.symponset.pval
```

```
## [1] 0.8125467
```

```
cex <- 0.9
cex_labels <- 0.9

{
  par(mfrow=c(1, 2), mai=c(0.5, 0.65, 0.1, 0.3), oma=c(1.75, 1, 0, 0))

  plot(peaktime/24 ~ p2, data=params.peaktime.data,
            col=colors[factor(dose)], xlim=c(-4, 1),
            ylim=c(1, 6), cex.axis=cex, cex=cex,
            pch=ifelse(delay < 0, 16, 1),
            xaxt="n",
            ylab = "",
            xlab = "",
         mgp=c(0, 0.8, 0), tck=-0.04, las=2, bty='l')
  axis(1, at=seq(-4, 1), labels=c(expression(10^-4),
                                  expression(10^-3),
                                  expression(10^-2),
                                  expression(10^-1),
                                  expression(10^0),
                                  expression(10^1)), cex.axis=cex,
       mgp=c(0, 0.8, 0), tck=-0.04)
  mtext(text="Day of peak shedding",
        side=2, line=2, cex=cex_labels)
  mtext(text=expression(Viral~replication~rate~(hour^-1)),
        side=1, line=0, cex=cex_labels, outer = TRUE, adj=0.52)
```

```
  abline(lm.fit.p2.peaktime, col="red")

  text(panellabelratio * 5 - 4, panellabelratio * 5 + 1, labels="A", cex=cex, font=2)

  plot(symp.onset ~ p2, data=params.symp.data,
    col=colors[factor(dose)], xlim=c(-4, 1),
             ylim=c(1, 4), cex.axis=cex, cex=cex,
             pch=ifelse(delay < 0, 16, 1),
             yaxt="n",
             xaxt="n",
             ylab = "",
             xlab = "", bty='l')

  axis(1, at=seq(-4, 1), labels=c(expression(10^-4),
                                  expression(10^-3),
                                  expression(10^-2),
                                  expression(10^-1),
                                  expression(10^0),
                                  expression(10^1)), cex.axis=cex,
       mgp=c(0, 0.8, 0), tck=-0.04)
  mtext(text="Day of symptom onset",
        side=2, line=2, cex=cex)

  axis(2, at=seq(1, 4), cex.axis=cex, mgp=c(0, 0.8, 0), tck=-0.04, las=2)

  abline(lm.fit.p2.symponset, col="red")

  text(panellabelratio * 5 - 4, panellabelratio * 3 + 1, labels="B", cex=cex, font=2)
}
```



The correlation between peak shedding and viral growth rate is verified by running a linear regression between the p2 and p3 parameters from the statistical model

```
lm.fit.p2.p3 <-
  lm(p3 ~ p2, data=participant.params)
p2.p3.pval <-
```

```
  summary(lm.fit.p2.p3)$coefficients[2,4]
p2.p3.pval
```

```
## [1] 0.001477611
```