Figures

Kristina Ceres

10/28/2020

load packages

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(ggplot2)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(Matrix)
```

```
##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(ggpubr)
library(rmarkdown)
library(matrixStats)
```

```
##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count
```

read in data

```
setwd("~/PhD/HMM_project/CT_HMM/Figures/")
## data from cows
data = read.csv("../data/hmm_ready_data.csv")

#model results
# 2 state model data
seed = 1585510210

scale_out2 <- read.csv(paste("../data/scale_out2",seed, ".csv", sep=""))
shape_out2 <- read.csv(paste("../data/shape_out2",seed, ".csv", sep=""))
qmat_out2 <- read.csv(paste("../data/qmat_out2",seed, ".csv", sep=""))
aic2 <- read.csv(paste("../data/aic_2",seed, ".csv", sep=""))
ip_out2 <- read.csv(paste("../data/ip_out2", seed, ".csv", sep=""))

# 3 state model data
scale_out3 <- read.csv(paste("../data/scale_out3", seed, ".csv", sep=""))
shape_out3 <- read.csv(paste("../data/shape_out3", seed, ".csv", sep=""))
qmat_out3 <- read.csv(paste("../data/qmat_out3", seed, ".csv", sep=""))
aic3 <- read.csv(paste("../data/aic_3", seed, ".csv", sep=""))
ip_out3 <- read.csv(paste("../data/ip_out3", seed, ".csv", sep=""))
```

Data transformations

Format data for 2 state model

```
# format data
get_result_df = function(df){
  df = as.matrix(df)
  df = t(df)
  colnames(df) = df[1,]
  df = df[-1, ]
  df = as_tibble(df)
  return(df)
}

#transform estimated parameters
# 2 states
scale_out2 = get_result_df(scale_out2)
shape_out2 = get_result_df(shape_out2)
qmat_out2 = get_result_df(qmat_out2)
aic2 = get_result_df(aic2)
ip_out2 = get_result_df(ip_out2)

outputs2 = bind_cols(ip_out2, shape_out2, scale_out2, qmat_out2, aic2)
```

```
## New names:
## * `0` -> `0...1`
## * `1` -> `1...2`
## * `0` -> `0...3`
```

```
## * ‘1‘ -> ‘1...4‘
## * ‘0‘ -> ‘0...5‘
## * ...
```

```r
colnames(outputs2) = c("ip_out0", "ip_out1", "shape_out0", "shape_out1",
                       "scale_out0", "scale_out1", "qout0", "qout1", "qout2", "qout3", "aic")
```

Format data for 3 state model

```r
scale_out3 = get_result_df(scale_out3)
shape_out3 = get_result_df(shape_out3)
qmat_out3 = get_result_df(qmat_out3)
aic3 = get_result_df(aic3)
ip_out3 = get_result_df(ip_out3)

outputs3 = bind_cols(ip_out3, shape_out3, scale_out3, qmat_out3, aic3)
```

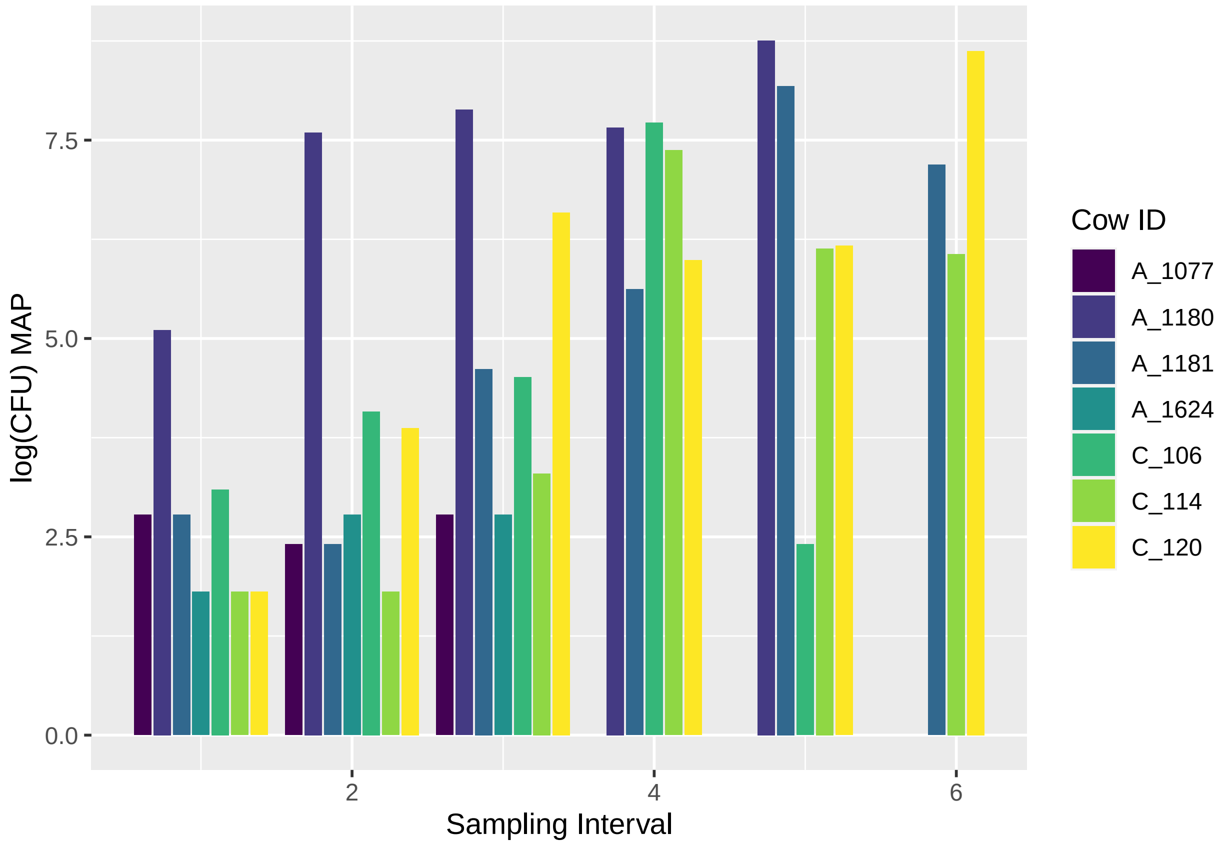
```
## New names:
## * ‘0‘ -> ‘0...1‘
## * ‘1‘ -> ‘1...2‘
## * ‘2‘ -> ‘2...3‘
## * ‘0‘ -> ‘0...4‘
## * ‘1‘ -> ‘1...5‘
## * ...
```

```r
colnames(outputs3) = c("ip_out0", "ip_out1", "ip_out2", "shape_out0", "shape_out1", "shape_out2", "scal
```

Figures

Figure 1

```r
data$SampDate = as.POSIXct(strptime(data$SampDate, format="%Y-%m-%d"))
p1data = data %>% group_by(CombinedID) %>% mutate(firstdate = min(SampDate)) %>%
  mutate(time2 =  difftime((SampDate), (firstdate), units = "weeks")) %>% mutate(SampTime = row_number(

idlist= c("A_1181", "C_106", "C_114", "A_1180", "A_1077", "A_1624", "C_120")
subset = p1data %>% filter(CombinedID %in% idlist)
p1 = ggplot(data = subset) +
  geom_col(aes(x=SampTime, y = cor_totCFU, group=CombinedID, fill=CombinedID), position=position_dodge2
  scale_fill_viridis(discrete=T)+labs(x= "Sampling Interval", y = "log(CFU) MAP", fill="Cow ID")

p1
```



Example gamma distributions used in Figure 2

```
shape3<- c(19.130472109595726, 16.774795300202804, 198.4831018750443)
scale3<- c(0.12568632084200168, 0.2815762121436809, 0.04040960031804022)
d0_ex =rgamma(n = 10000, shape=shape3[1], scale=scale3[1])
d1_ex =rgamma(n = 10000, shape=shape3[2], scale=scale3[2])
d2_ex =rgamma(n = 10000, shape=shape3[3], scale=scale3[3])
d0_ex =dgamma(seq(1,12,length=1000), shape=shape3[1], scale=scale3[1])
d1_ex =dgamma(seq(1,12,length=1000), shape=shape3[2], scale=scale3[2])
d2_ex =dgamma(seq(1,12,length=1000), shape=shape3[3], scale=scale3[3])
gam_ex  = tibble(d0_ex, d1_ex, d2_ex, x=seq(1,12,length=1000))
gam_ex = gam_ex %>% gather(key = "distribution", value = "density",-x)

p2 = ggplot()+
  geom_line(data = gam_ex, aes(x=x, y=density, color = distribution)) +
  labs(x = "log(CFU) Value")+
  scale_color_manual(values=c("#874E9A","#50AAA3","#BAC438"), labels=c("State 0", "State 1", "State 2")
  scale_x_continuous(limits=c(.5,10))+theme_grey()
p2
```

```
## Warning: Removed 546 row(s) containing missing values (geom_path).
```

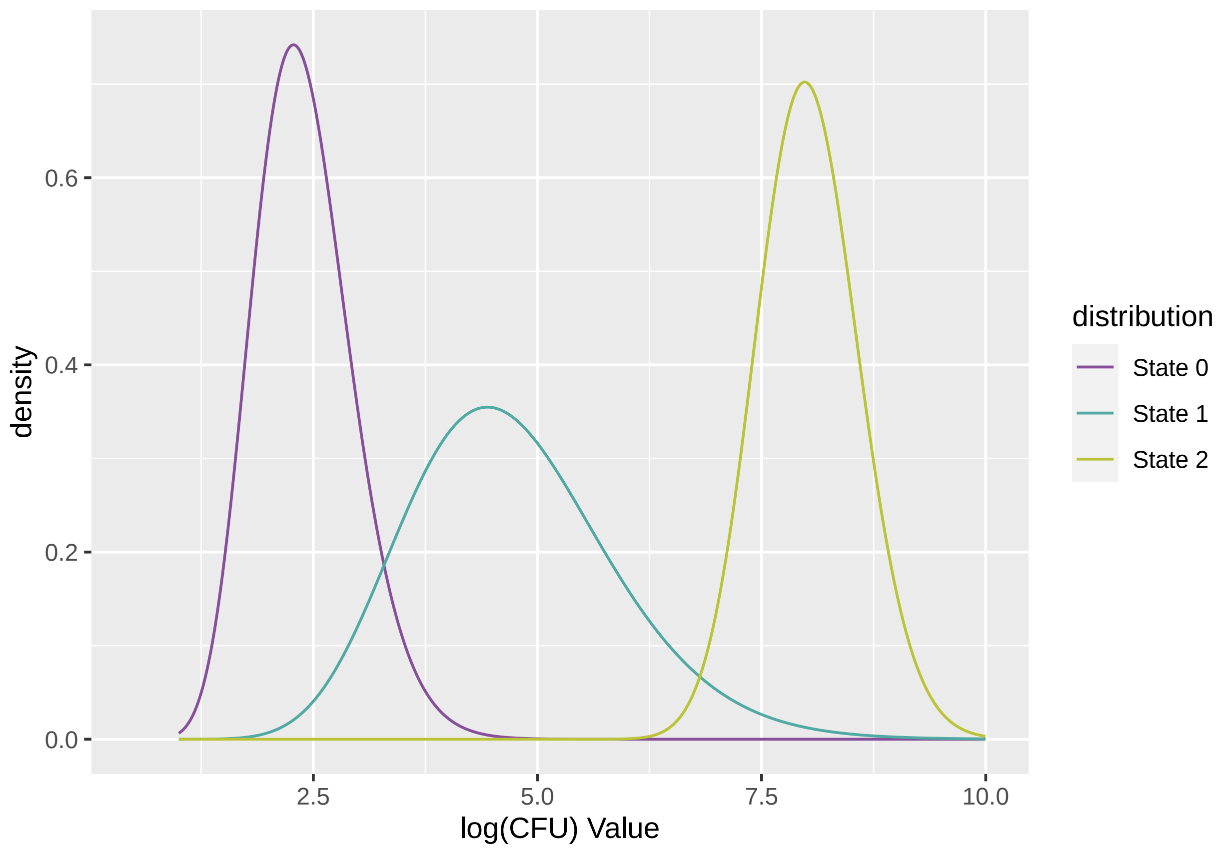


Figure 3

Formatting data for transient distributiions

```
n=nrow(data)
outputs2 = outputs2 %>% filter(aic>0) %>% arrange(aic)
outputs3 = outputs3 %>% filter(aic > 0) %>% arrange(aic)
ip2 = c(outputs2[1,]$ip_out0, outputs2[1,]$ip_out1)
qmat2 = matrix(c(outputs2[1,]$qout0,outputs2[1,]$qout1,
                 outputs2[1,]$qout2,outputs2[1,]$qout3), nrow=2, byrow=T)


ip3 = c(outputs3[1,]$ip_out0, outputs3[1,]$ip_out1, outputs3[1,]$ip_out2)
qmat3 = matrix(c(outputs3[1,]$qout0,outputs3[1,]$qout1,outputs3[1,]$qout2,
                 outputs3[1,]$qout3,outputs3[1,]$qout4,outputs3[1,]$qout5,
                 outputs3[1,]$qout6,outputs3[1,]$qout7,outputs3[1,]$qout8), nrow=3, byrow=T)

n = 300
# make data frames
q2 = tibble(x = seq(1,n, length=n), "0" = rep(0, n), "1" = rep(0,n), group = rep("2 State Model", n))
q3 = tibble(x = seq(1,n, length=n), "0" = rep(0, n), "1"= rep(0,n), "2" = rep(0,n), group = rep("3 Stat

# get state probabilities for each position in the chain
for (i in 1:n){
  temp2 = ip2 %*% expm(qmat2 * q2$x[i])
```

```
  q2$`0`[i]=temp2[1]
  q2$`1`[i]=temp2[2]

  temp3 = ip3 %*% expm(qmat3 * q3$x[i])
  q3$`0`[i]=temp3[1]
  q3$`1`[i]=temp3[2]
  q3$`2`[i]=temp3[3]

}

q2 = gather(q2, -c(x, group), key= "State", value="Probability")
q3 = gather(q3, -c(x, group), key= "State", value="Probability")

q = bind_rows(q2, q3)
```

Evaluating stationary distributions

```
#stationary distribution pi, where pi * Q = 0, and Q is the transition rate matrix
#2 state model
q2 <- matrix(NA, nrow = 2, ncol = 2)
q2[1,] <- t(qmat2)[1,]
q2[2,] <- c(1,1)
pi2 <- solve(q2,c(0,1))

#3 state model
q3 <- matrix(NA, nrow = 3, ncol = 3)
q3[1:2,] <- t(qmat3)[1:2,]
q3[3,] <- c(1,1,1)
pi3 <- solve(q3,c(0,0,1))
```

Generate a dataframe of samples from gamma distributions with fitted shape and scale parameters

```
n = nrow(data)
shape2 <- c(outputs2[1,]$shape_out0, outputs2[1,]$shape_out1)
scale2 <- c(outputs2[1,]$scale_out0, outputs2[1,]$scale_out1)

d02 =dgamma(seq(1,12,length=1000), shape=shape2[1], scale=scale2[1])
d12 =dgamma(seq(1,12,length=1000), shape=shape2[2], scale=scale2[2])

shape3 <- c(outputs3[1,]$shape_out0, outputs3[1,]$shape_out1, outputs3[1,]$shape_out2)
scale3 <- c(outputs3[1,]$scale_out0, outputs3[1,]$scale_out1, outputs3[1,]$scale_out2)

d03 =dgamma(seq(1,12,length=1000), shape=shape3[1], scale=scale3[1])
d13 =dgamma(seq(1,12,length=1000), shape=shape3[2], scale=scale3[2])
d23 =dgamma(seq(1,12,length=1000), shape=shape3[3], scale=scale3[3])

gammas20 = tibble(dist = d02, State = rep("0", length(d02)), group = rep("2 State Model", length(d02)),
gammas21 = tibble(dist = d12, State = rep("1", length(d12)), group = rep("2 State Model", length(d12)),

gammas30 = tibble(dist = d03, State = rep("0", length(d03)), group = rep("3 State Model", length(d03)),
gammas31 = tibble(dist = d13, State = rep("1", length(d13)), group = rep("3 State Model", length(d13)),
gammas32 = tibble(dist = d23, State = rep("2", length(d23)), group = rep("3 State Model", length(d23)),
```

```
gammas = bind_rows(gammas20, gammas21, gammas30, gammas31, gammas32)
```

Generate subplots for Figure 3 and stich them together

```
p3a = ggplot(data = q)+geom_line(aes(x=x, y=Probability, color=State))+facet_wrap(~group, nrow=1)+
  labs(x = "t", y =expression(pi[0]*e^(Qt)))+
  scale_color_manual(values=c("#874E9A","#50AAA3","#BAC438")) + theme_gray()

p3.1 = ggplot()+
  stat_density(data = gammas, aes(x=dist, group = State, fill = State),position="identity", alpha = .7)
  facet_wrap(~group, nrow=1)+
  scale_fill_manual(values=c("#874E9A","#50AAA3","#BAC438"))+
  labs(x= "log(CFU) MAP", y = "Density") + theme_grey()
p3.1 = ggplot()+
  geom_line(data = gammas, aes(x=x, y=dist,group = State, color = State)) +
  facet_wrap(~group, nrow=1)+
  scale_color_manual(values=c("#874E9A","#50AAA3","#BAC438"))+
  labs(x= "log(CFU) MAP", y = "Density") + theme_grey()

p3.2 <- ggplot()+
  geom_histogram(data = data, aes(x=cor_totCFU), binwidth=.2, alpha = .7)+
  labs(x= "log(CFU) MAP", y = "Number of observations") + theme_grey()

p3.BC<- ggarrange(p3.2, p3.1,
                  labels=c("B","C"),
                  ncol = 2, nrow = 1,
                  widths = c(.3, .7))

p3 <- ggarrange(p3a, p3.BC,
                labels=c("A"),
                ncol=1, nrow=2,
                heights=c(.5,.5))
p3
```

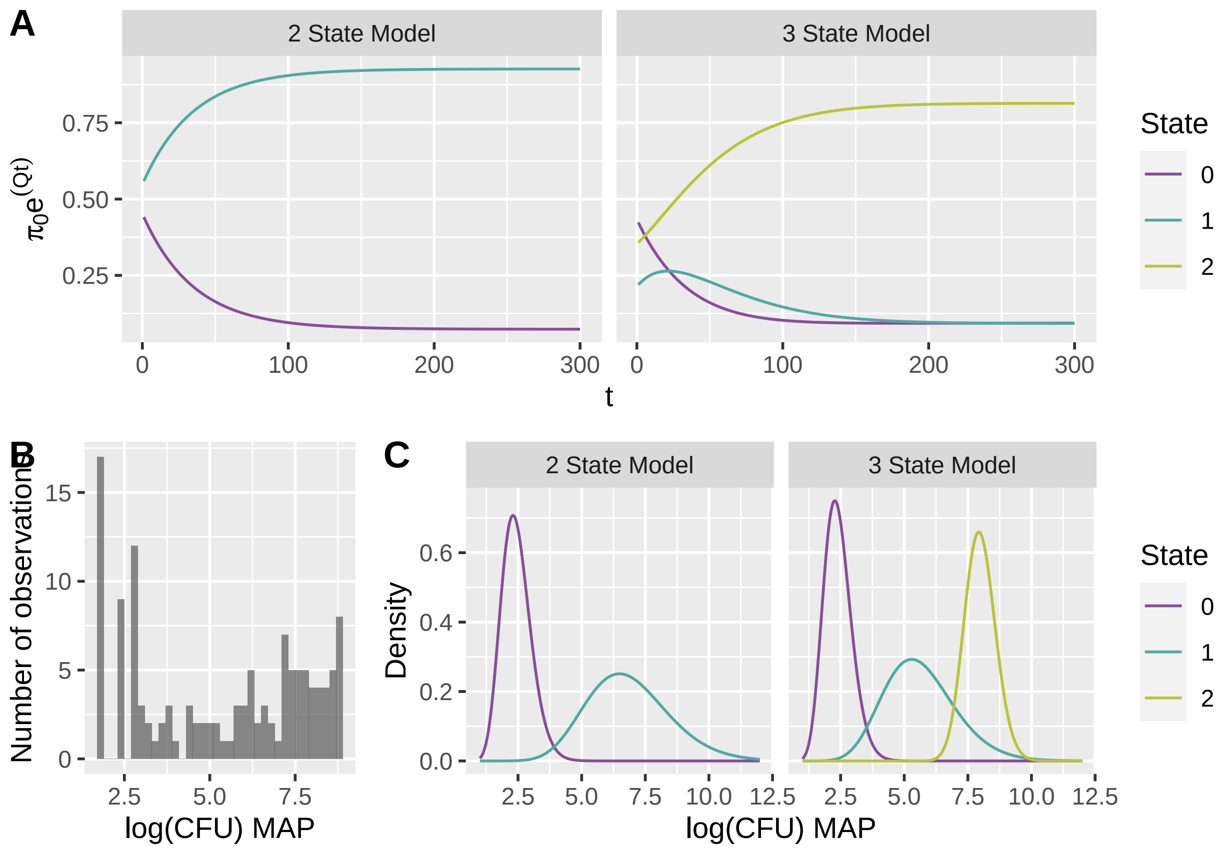


Figure 4

Load posterior probability data and transform

```
read_posterior = function(filename, time_data, nstates, state){
  txt <- gsub("\\[|\\]", "", readLines(filename))
  post = read.csv(text=txt)
  colnames(post) = c("CowID", 1,2,3,4,5,6,7,8,9,10,11)
  post = gather(post, key="SampleTime", value = "prob", -c(CowID)) %>% filter(is.na(prob)==F)
  post = post %>% arrange(CowID)
  p = bind_cols(post, time_data)
  p$nstates = as.factor(rep(nstates, length=nrow(p)))
  p$state = as.factor(rep(state, length=nrow(p)))
  return(p)
}
```

Load posterior probability data and transform

```
time_data = data %>% mutate(time = ifelse(is.na(time) == T, 0, time)) %>%
  group_by(CombinedID) %>% mutate(times = cumsum(time)) %>% select(CombinedID, times) %>%
  rename(CowID = CombinedID) %>% arrange(CowID)


p0_2 = read_posterior("../data/posterior0_df_2.csv", time_data, 2, 0)
```

```
## New names:
## * CowID -> CowID...1
## * CowID -> CowID...4
```

```r
p1_2 = read_posterior("../data/posterior1_df_2.csv", time_data, 2, 1)
```

```
## New names:
## * CowID -> CowID...1
## * CowID -> CowID...4
```

```r
p2 = bind_rows(p0_2, p1_2)
p2 = p2  %>% group_by(CowID...1, state)  %>%
  mutate(Endstate = case_when(
    state==0 & last(prob) > 0.75 ~ "0",
    state== 1 & last(prob) > 0.75 ~ "1",
    state== 0 & last(prob) < 0.75 ~ "1",
    state== 1 & last(prob) < 0.75 ~ "0"
  ))

p0_3 = read_posterior("../data/posterior0_df_3.csv", time_data, 3, 0)
```

```
## New names:
## * CowID -> CowID...1
## * CowID -> CowID...4
```

```r
p1_3 = read_posterior("../data/posterior1_df_3.csv", time_data, 3, 1)
```

```
## New names:
## * CowID -> CowID...1
## * CowID -> CowID...4
```

```r
p2_3 = read_posterior("../data/posterior2_df_3.csv", time_data, 3, 2)
```

```
## New names:
## * CowID -> CowID...1
## * CowID -> CowID...4
```

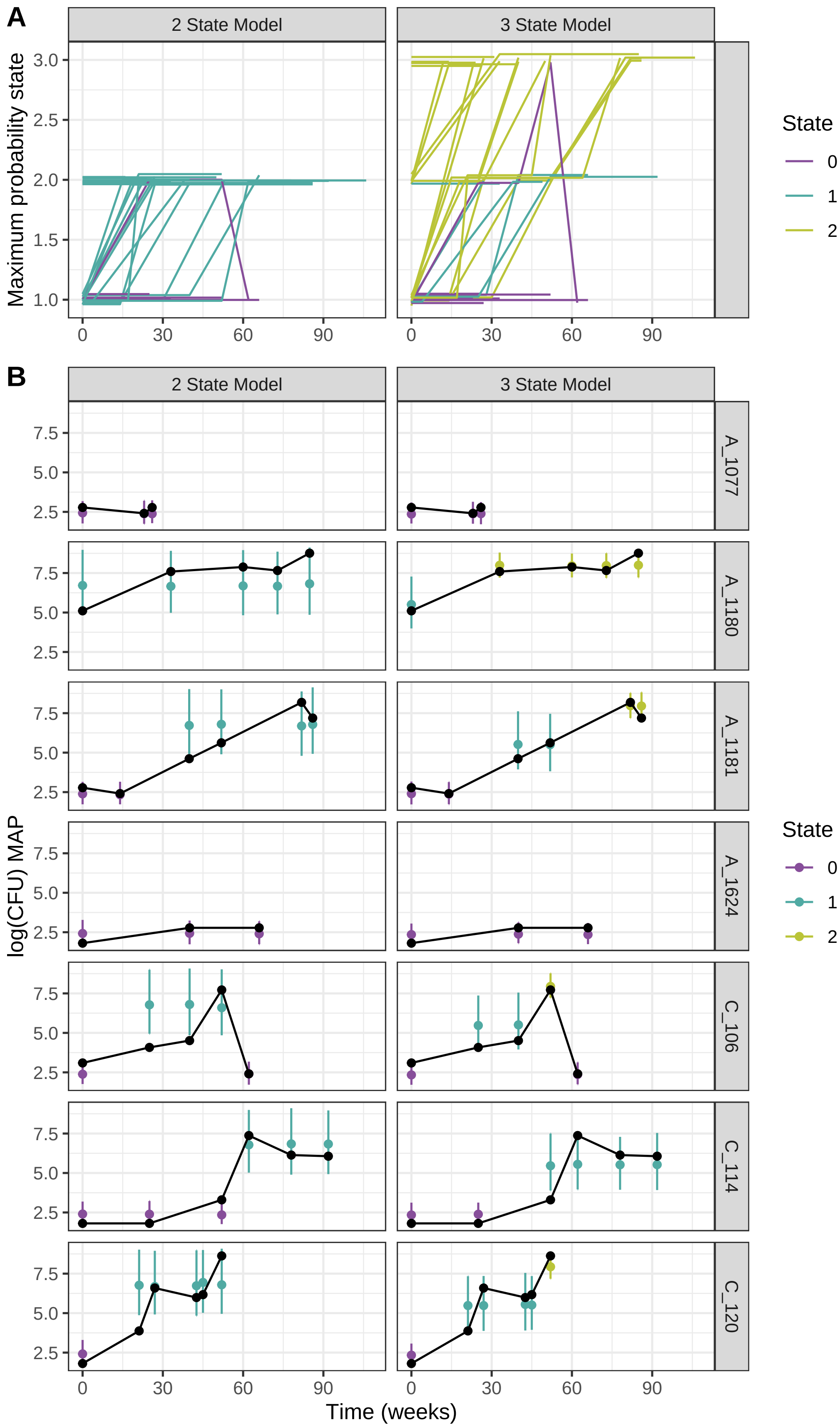
```r
p3 = bind_rows(p0_3, p1_3, p2_3)
p3 = p3 %>%  group_by(CowID...1, state) %>% mutate(max_last_prob = max(last(prob))) %>%
  group_by(CowID...1) %>% mutate(max_end_prob = max(max_last_prob)) %>%
  group_by(CowID...1, state)  %>%
  mutate(Endstate = case_when(
    state==0 & last(prob) == max_end_prob ~ "0",
    state==1 & last(prob) == max_end_prob ~ "1",
    state==2 & last(prob) == max_end_prob ~ "2")) %>%
  group_by(CowID...1) %>%
  mutate(Endstate = case_when(
    "0" %in% Endstate ~ "0",
    "1" %in% Endstate ~ "1",
    "2" %in% Endstate ~ "2"
```

```
  ))

post= bind_rows(p2,p3)

post = post %>% group_by(CowID...1, nstates, times) %>% mutate(max_prob = ifelse(prob == max(prob), sta
post = post %>% group_by(CowID...1, nstates, times) %>%
  mutate(max_prob = ifelse(max_prob == max(max_prob), max_prob, max(max_prob)))

post = post %>% group_by(CowID...1, nstates)%>% mutate(max_prob = as.numeric(max_prob)+runif(1, -.05,.05
  mutate(nstates2 = ifelse(nstates == "2" , "2 State Model", "3 State Model"))

post2 <- post %>% group_by(nstates, CowID...1, SampleTime) %>% mutate(max = max(prob)) %>% filter(prob==
p1data$SampleTime <- as.character(p1data$SampTime)
p1data$CowID...1 <- p1data$CombinedID
post_w_cfu <- left_join(post2, p1data, by=c("CowID...1", "SampleTime"))

post_w_cfu$placeholder = " "

ndraws=1000
rand_cfus <- matrix(data=NA, nrow = nrow(post_w_cfu), ncol=ndraws)
for (r in 1:nrow(post_w_cfu)){
  if(post_w_cfu$nstates[r] == 2 & post_w_cfu$state[r] == 0){
    rand_cfus[r,]= rgamma(ndraws, shape=shape2[1], scale=scale2[1])
  }
   if(post_w_cfu$nstates[r] == 2 & post_w_cfu$state[r] == 1){
     rand_cfus[r,]= rgamma(ndraws, shape=shape2[2], scale=scale2[2])
   }
   if(post_w_cfu$nstates[r] == 3 & post_w_cfu$state[r] == 0){
     rand_cfus[r,]= rgamma(ndraws, shape=shape3[1], scale=scale3[1])
   }
   if(post_w_cfu$nstates[r] == 3 & post_w_cfu$state[r] == 1){
     rand_cfus[r,]= rgamma(ndraws, shape=shape3[2], scale=scale3[2])
   }
   if(post_w_cfu$nstates[r] == 3 & post_w_cfu$state[r] == 2){
     rand_cfus[r,]= rgamma(ndraws, shape=shape3[3], scale=scale3[3])
   }
}
probs=c(.1, .5, .9)
summary_df <- as_tibble(rowQuantiles(rand_cfus, probs=probs))
colnames(summary_df) = c("ten", "median", "ninety")

new <- bind_cols(post_w_cfu, summary_df)
subset2 = new %>% filter(CombinedID %in% idlist)
```

Create subplots and combine

```
p4a = ggplot(post_w_cfu)+geom_line(aes(x=times, y = max_prob, group=CowID...1, color=Endstate)) +
  facet_grid(cols=vars(nstates2), rows=vars(placeholder))+
  scale_color_manual(values=c("#874E9A","#50AAA3","#BAC438"))+
  labs(color = "State", x = NULL, y = "Maximum probability state")+ theme_bw()+
  scale_x_continuous(limits=c(0,108))
```

```
p4b <- ggplot(subset2)+
  geom_point(aes(x=time2, y=median, color=state))+
  geom_errorbar(aes(x=time2, ymin=ten, ymax=ninety, color=state))+
  geom_line(aes(x=time2, y = cor_totCFU), color = "black")+
  geom_point(aes(x=time2, y= cor_totCFU), color = "black")+
  labs(x= "Time (weeks)", y = "log(CFU) MAP", color="State")+
  scale_color_manual(values=c("#874E9A","#50AAA3","#BAC438"))+
  theme(legend.position="right")+
  facet_grid(cols=vars(nstates2), rows=vars(CowID...1))+
  scale_x_continuous(limits=c(0,108))+theme_bw()


p4 <- ggarrange(p4a, p4b,
                labels = c("A", "B"),
                ncol = 1, nrow = 2,
                heights = c(.25,.75))
p4
```



A
2 State Model
3 State Model
Maximum probability state
State
0
1
2
B
A_1077
A_1180
A_1181
A_1624
C_106
C_114
C_120
log(CFU) MAP
Time (weeks)