

# Figures

Kristina Ceres

7/30/2020

## load packages

```
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
```

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

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

```
library(Matrix)
```

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

```
library(ggpubr)
```

## read in data

```

## 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)
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

```

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

```

## Figures

Figure 1

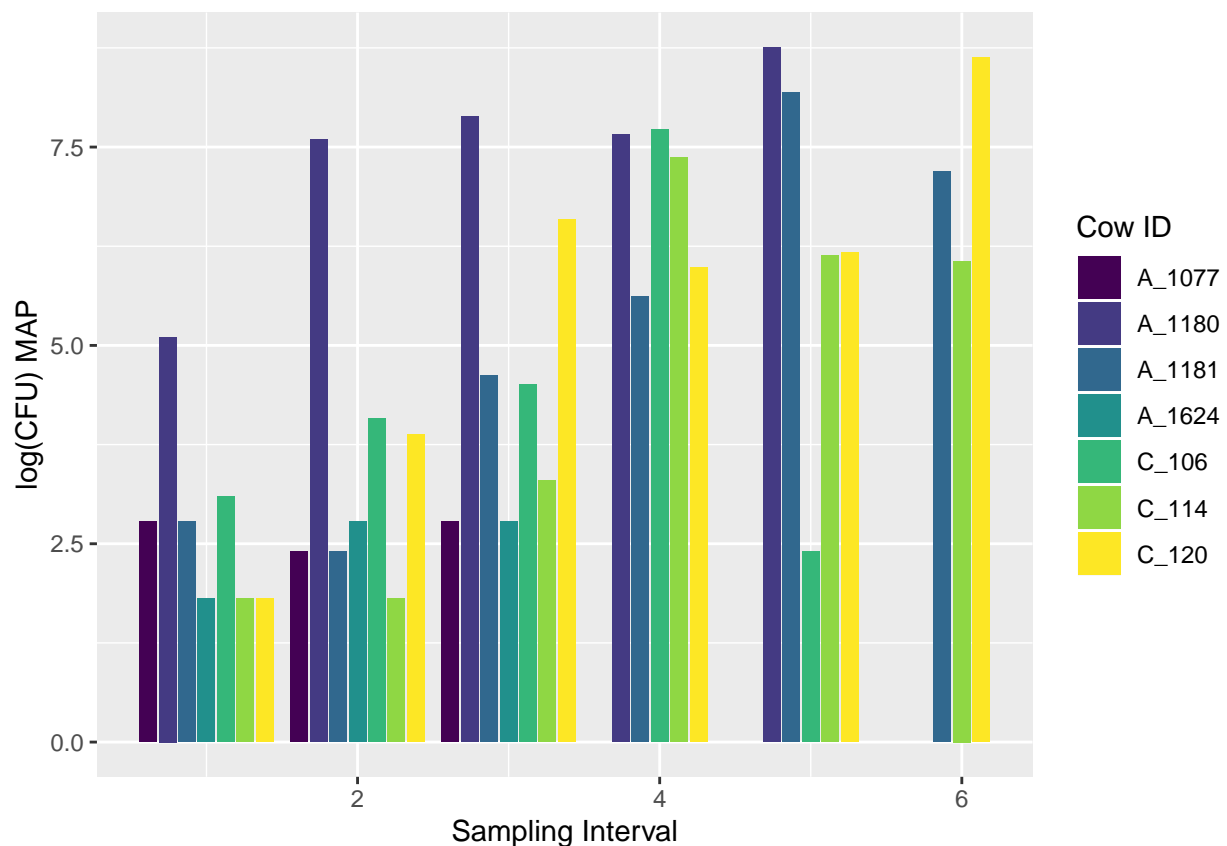
```

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")

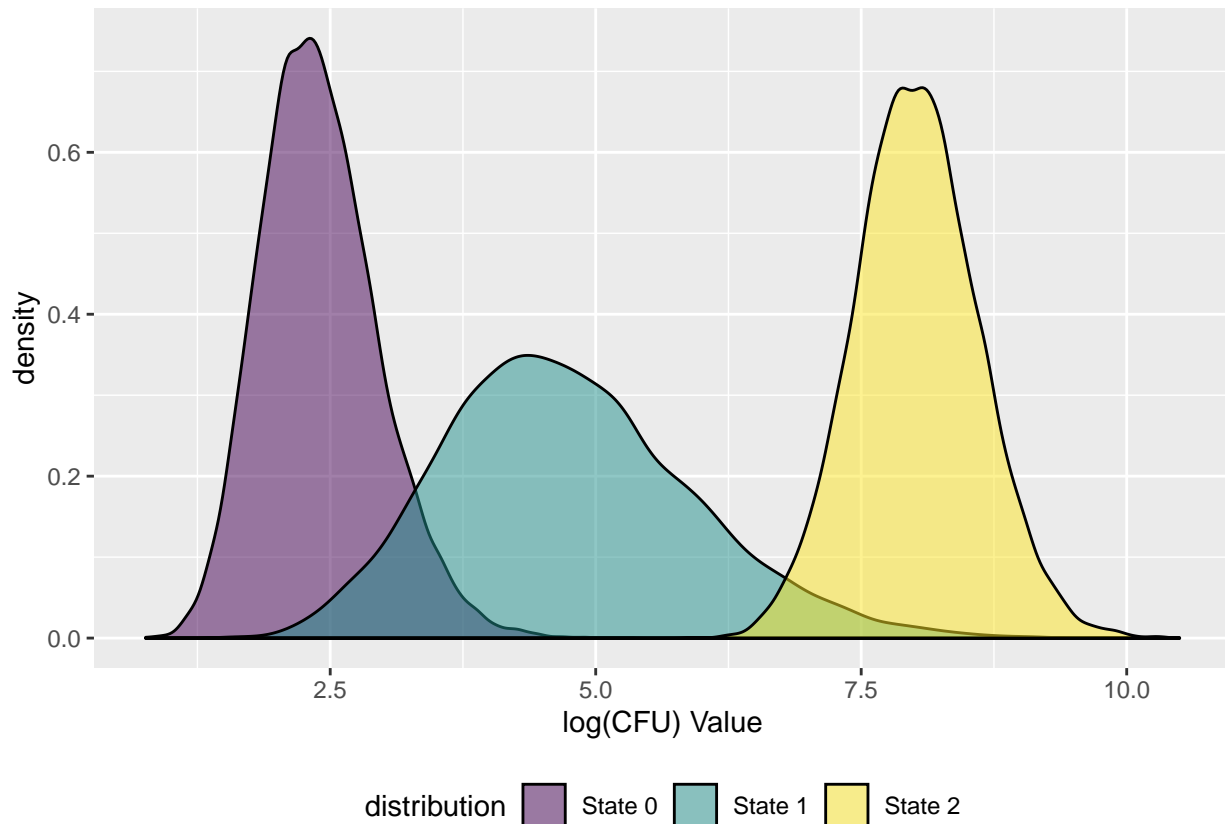
p1

```



## 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])
gam_ex = tibble(d0_ex, d1_ex, d2_ex)
gam_ex = gam_ex %>% gather(key = "distribution", value = "density")
p2 = ggplot()+
  geom_density(data = gam_ex, aes(density, fill = distribution), alpha = .5) +
  labs(x = "log(CFU) Value")+ scale_fill_viridis(discrete=T, labels=c("State 0", "State 1", "State 2"))+
  theme(legend.position="bottom")
p2
```



**Figure 3**

Formatting data for transient distributions

```
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", n))
q3 = tibble(x = seq(1,n, length=n), "0" = rep(0, n), "1" = rep(0,n), "2" = rep(0,n), group = rep("3 State", n))

# 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 =rgamma(n = round(pi2[1]*n), shape=shape2[1], scale=scale2[1])

```

```

d12 =rgamma(n = round(pi2[2]*n), 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 =rgamma(n = round(pi3[1] * n), shape=shape3[1], scale=scale3[1])
d13 =rgamma(n = round(pi3[2] * n), shape=shape3[2], scale=scale3[2])
d23 =rgamma(n = round(pi3[3] * n), shape=shape3[3], scale=scale3[3])

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

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

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

q$type = rep("stationary", nrow(q))
gammas$type = rep("emiss", nrow(gammas))

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

```

Generate subplots for Figure 3 and stitch 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_grey()

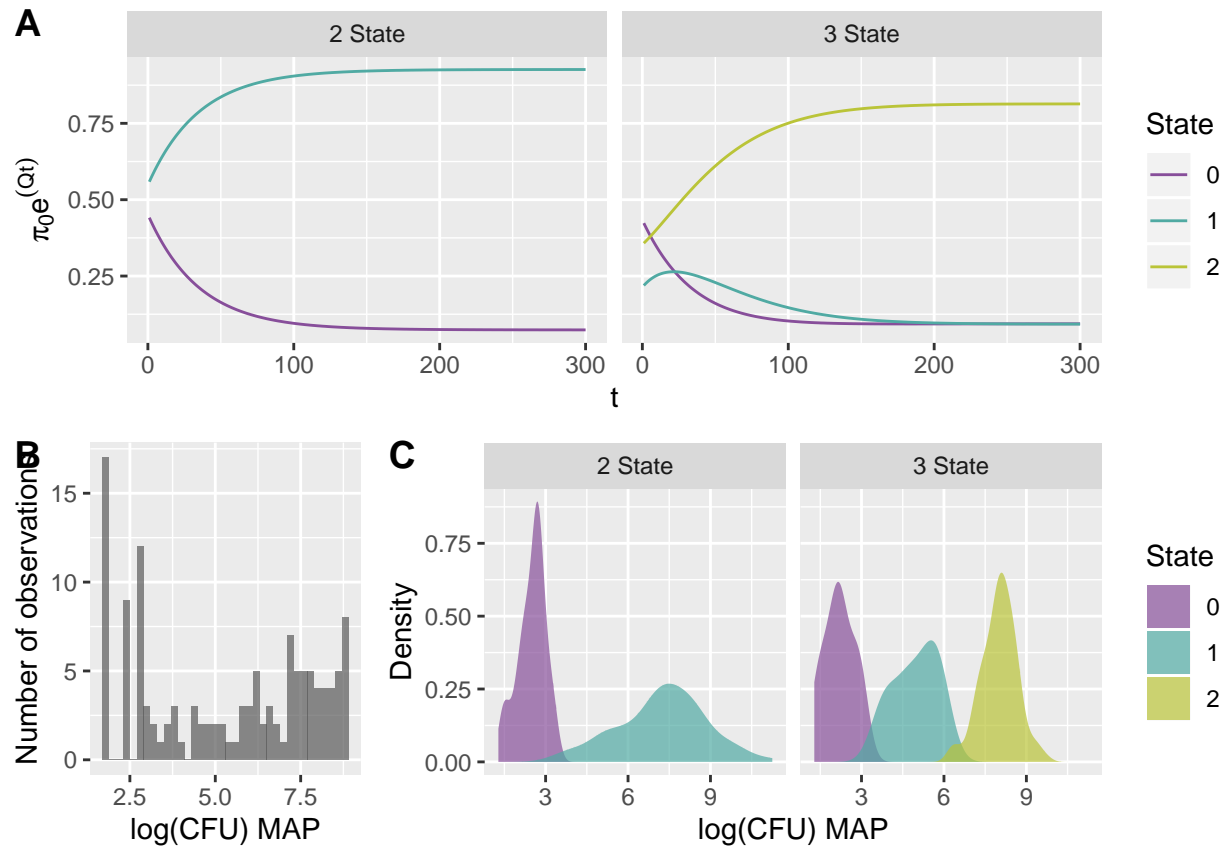
p3.1 = ggplot()+
  stat_density(data = gammas, aes(x=dist, group = State, fill = State),position="identity", alpha = .7)+
  facet_wrap(~group, nrow=1)+
  scale_fill_manual(values=c("#874E9A", "#50AAA3", "#BAC438"))+
  labs(x= "log(CFU) MAP", y = "Density") + theme_grey()

p3.2 <- 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)
```

```
p1_2 = read_posterior("../data/posterior1_df_2.csv", time_data, 2, 1)
p2 = bind_rows(p0_2, p1_2)
```

```
## Warning in bind_rows(x, .id): Unequal factor levels: coercing to character
```

```
## Warning in bind_rows(x, .id): binding character and factor vector, coercing
## into character vector
```

```
## Warning in bind_rows(x, .id): binding character and factor vector, coercing
## into character vector
```

```
p2 = p2 %>% group_by(CowID, 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)
p1_3 = read_posterior("../data/posterior1_df_3.csv", time_data, 3, 1)
p2_3 = read_posterior("../data/posterior2_df_3.csv", time_data, 3, 2)
p3 = bind_rows(p0_3, p1_3, p2_3)
```

```
## Warning in bind_rows(x, .id): Unequal factor levels: coercing to character
```

```
## Warning in bind_rows(x, .id): binding character and factor vector, coercing
## into character vector
```

```
## Warning in bind_rows(x, .id): binding character and factor vector, coercing
## into character vector
```

```
## Warning in bind_rows(x, .id): binding character and factor vector, coercing
## into character vector
```

```
p3 = p3 %>% group_by(CowID, state) %>% mutate(max_last_prob = max(last(prob))) %>%
  group_by(CowID) %>% mutate(max_end_prob = max(max_last_prob)) %>%
  group_by(CowID, 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) %>%
  mutate(Endstate = case_when(
    "0" %in% Endstate ~ "0",
    "1" %in% Endstate ~ "1",
    "2" %in% Endstate ~ "2"
  ))
```

```
post= bind_rows(p2,p3)
```



```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
## Warning in bind_rows_(x, .id): binding character and factor vector, coercing
## into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector, coercing
## into character vector
```

```
post = post %>% group_by(CowID, nstates, times) %>% mutate(max_prob = ifelse(prob == max(prob), state,
post = post %>% group_by(CowID, nstates, times) %>%
  mutate(max_prob = ifelse(max_prob == max(max_prob), max_prob, max(max_prob)))

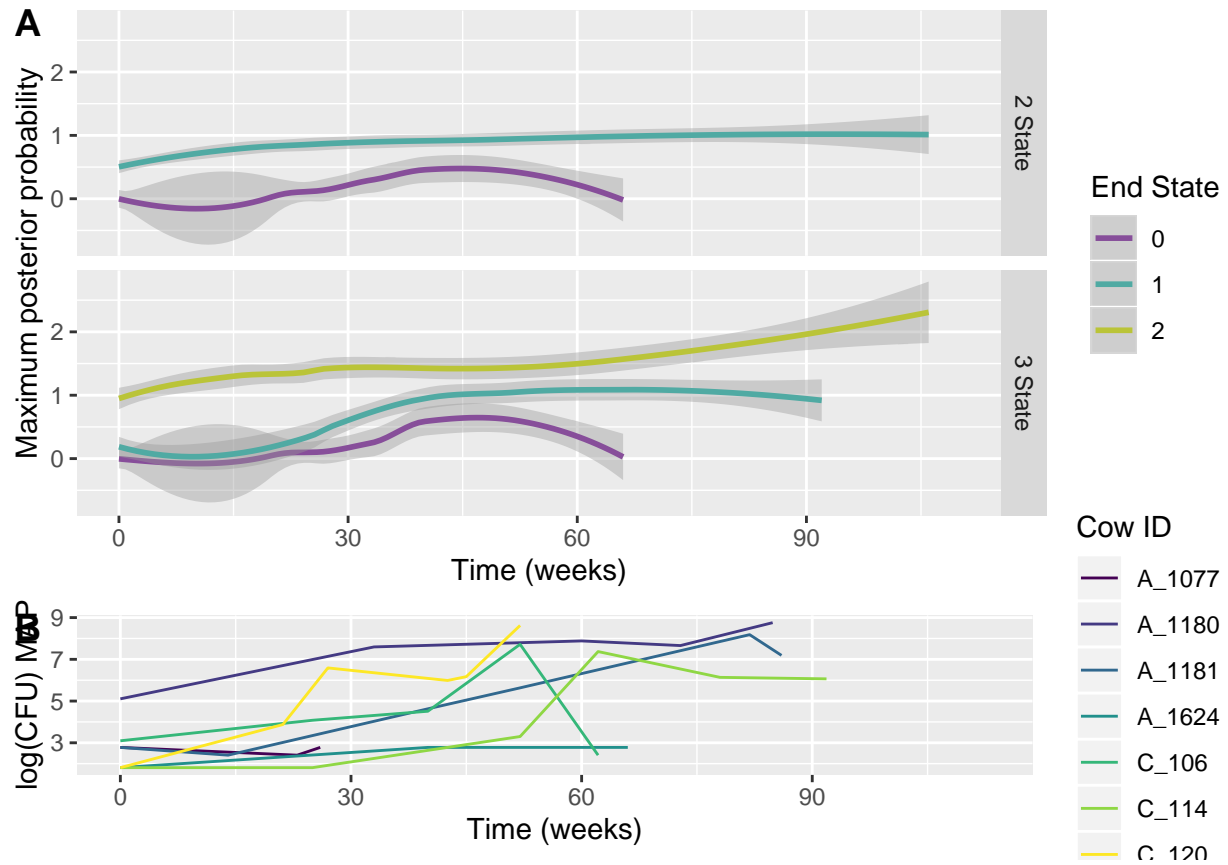
post = post %>% group_by(CowID, nstates)%>% mutate(max_prob = as.numeric(max_prob)+runif(1, -.05,.05)) %
  mutate(nstates2 = ifelse(nstates == "2" , "2 State", "3 State"))
```

Create subplots and combine

```
p4a <- ggplot(post, aes(x = times, y = max_prob, group=Endstate, color=Endstate)) +
  facet_grid(rows=vars(nstates2))+ scale_color_manual(values=c("#874E9A", "#50AAA3", "#BAC438"))+
  theme(legend.position="right")+labs(x="Time (weeks)", y = "Maximum posterior probability", color = "E

p4b <- ggplot(data = subset) +
  geom_line(aes(x=time2, y = cor_totCFU, group=CombinedID, color=CombinedID))+
  scale_color_viridis(discrete=T)+labs(x= "Time (weeks)", y = "log(CFU) MAP", color="Cow ID")+
  theme(legend.position="right")+scale_x_continuous(limits=c(0,113))+ facet_grid()

p4 <- ggarrange(p4a, p4b,
  labels = c("A", "B"),
  ncol = 1, nrow = 2,
  heights = c(.7,.3))
p4
```



#Supplemental Figure 1

```
s1 = ggplot(post)+geom_line(aes(x=times, y = max_prob, group=CowID, color=Endstate)) +
  facet_grid(vars(nstates2))+ scale_color_manual(values=c("#874E9A", "#50AAA3", "#BAC438"))+
  labs(color = "End State", x = "Time", y = "Maximum posterior probability")+ theme_grey()
s1
```

