A stochastic model for the transmission of COVID-19 in Georgia

March 14, 2020

Introduction

The epidemiology of COVID-19 in the United States is poorly understood. Here we develop a model to understand transmission in the state of Georgia. Key features of this model include:

1. Stochastic transmission process
2. Realistic wait time distributions
3. Time varying rates of case detection, isolation, and case notification

This model was parameterized using clinical outcome reports from the epidemic in Hubei province, China and further calibrated by fitting to case notification data. The COVID-19 epidemic is changing rapidly and information that was used in the construction of this model may be incomplete or contain errors. Accordingly, these results are preliminary, provisional, and subject to change. These results have not been peer-reviewed, but have been prepared to a professional standard with the intention of providing useful interpretation of a rapidly developing event.

Simulation functions

Here we provide functions for evaluating the model.

The function `onestep` simulates one time step in the transmission process.

```
onestep <- function (x, params) {  #function to calculate one step of stochastic SIR

  S <- x[2]                             #local variable for susceptibles

  E1 <- x[3]                            #exposed classes
  E2 <- x[4]
  E3 <- x[5]
  E4 <- x[6]
  E5 <- x[7]
  E6 <- x[8]

  I1 <- x[9]                            #detected infectious classes
  I2 <- x[10]
  I3 <- x[11]
  I4 <- x[12]

  Iu1 <- x[13]                            #undetected infectious classes
  Iu2 <- x[14]
  Iu3 <- x[15]
  Iu4 <- x[16]

  I.detected <-I1+I2+I3+I4
  I.undetected <- Iu1+Iu2+Iu3+Iu4
  I <- I1+I2+I3+I4+Iu1+Iu2+Iu3+Iu4      #total infections
```

```
H <- x[17]                           #local variable for hospitalized
Ru <- x[18]                          #local variable for undetected recovereds

C <- x[19]    # local variable for notifications

N <- S+E1+E2+E3+E4+E5+E6+I1+I2+I3+I4+Iu1+Iu2+Iu3+Iu4+H+Ru          # total size of population

t <- x[20]                          #get current time


with(                                #use with to simplify code
     as.list(params),
     {
       gammai <- 4*gamma(z=z, b=b, a0=a0, t=as.numeric(t))  # multiplier 4 for pseudo stages
       sigmai <- 6*sigma  # multiplier 6 for pseudo stages
       etat <- eta(t)
       betat <- beta(t)

       rates <- as.numeric(c(betat*I.detected/N+betat*c*I.undetected/N,                           # m
                  sigmai, sigmai, sigmai, sigmai, sigmai, sigmai,   # movements out of E
                  gammai, gammai, gammai, gammai,                   # movements out of I (detected)
                  b, b, b, b,                                       # movements out of I (undetected)
                  etat))

       states0 <- x[2:(length(x)-1)]

       # transition probabilities

       p <- matrix(0, nrow=length(rates),ncol=length(states0))                                     #

       p[1,]  <- c(exp(-rates[1]*dt),1-exp(-rates[1]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

       p[2,]  <- c(0, exp(-rates[2]*dt), 1-exp(-rates[2]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       p[3,]  <- c(0, 0, exp(-rates[3]*dt), 1-exp(-rates[3]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       p[4,]  <- c(0, 0, 0, exp(-rates[4]*dt), 1-exp(-rates[4]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       p[5,]  <- c(0, 0, 0, 0, exp(-rates[5]*dt), 1-exp(-rates[5]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       p[6,]  <- c(0, 0, 0, 0, 0, exp(-rates[6]*dt), 1-exp(-rates[6]*dt), 0, 0, 0, 0, 0, 0, 0, 0, 0,
       p[7,]  <- c(0, 0, 0, 0, 0, 0, exp(-rates[7]*dt), (1-exp(-rates[7]*dt))*q(t,w), 0, 0, 0, (1-exp

       p[8,]  <- c(0, 0, 0, 0, 0, 0, 0, exp(-rates[8]*dt), 1-exp(-rates[8]*dt), 0, 0, 0, 0, 0, 0, 0,
       p[9,]  <- c(0, 0, 0, 0, 0, 0, 0, 0, exp(-rates[9]*dt), 1-exp(-rates[9]*dt), 0, 0, 0, 0, 0, 0,
       p[10,] <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, exp(-rates[10]*dt), 1-exp(-rates[10]*dt), 0, 0, 0, 0, 0
       p[11,] <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, exp(-rates[11]*dt), 0, 0, 0, 0, 1-exp(-rates[11]*dt)

       p[12,]  <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  exp(-rates[12]*dt), 1-exp(-rates[12]*dt), 0, 0,
       p[13,]  <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  exp(-rates[13]*dt), 1-exp(-rates[13]*dt), 0,
       p[14,] <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  exp(-rates[14]*dt), 1-exp(-rates[14]*dt),
       p[15,] <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  exp(-rates[15]*dt), 0, 1-exp(-rates[15]


       p[16,] <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, exp(-rates[16]*dt), 0, 1-exp(-rates[1

       # update states
```

```
        states1 <- matrix(0, nrow=length(rates),ncol=length(states0))                   #

        for(i in 1:length(rates)){
          states1[i,] <- t(rmultinom(1, states0[i], p[i,]))
        }

        states1 <- colSums(states1)
        states1[17] <- states1[17]+Ru  #add formerly Recovered undetected cases
        states1[18] <- states1[18]+C  #add formerly notified cases

        return(x <- c(dt, states1, tail(x,1)+dt))
      }
      )
}
```

The function `model` iteratively applies `onestep` to generate a solution of the stochastic model.

```
model <- function (x, params, nstep) {  #function to simulate stochastic SIR
  output <- array(dim=c(nstep+1,length(x)))         #set up array to store results
  colnames(output) <- c("time","S",
                         "E1", "E2", "E3", "E4", "E5", "E6",
                         "I1", "I2", "I3", "I4", "Iu1", "Iu2", "Iu3", "Iu4",
                         "H", "Ru", "C", "cum.time") #name variables
  output[1,] <- x                           #first record of output is initial condition
  for (k in 1:nstep) {                      #iterate for nstep steps
    output[k+1,] <- x <- as.numeric(onestep(x,params))
  }
  output                                    #return output
}
```

The function `evaluate.model` simulates an arbitrary number of realizations and returns the result.

With 10.52 million people, the population of Georgia is about the same as the city of Wuhan, China. We will initialize the model on March 1.

It is difficult to know how to initialize the model since testing in the state has been slow. Numerous, seemingly unlinked cases were identified in the first week of March and the first death occurred on 12 March. We attemt to put bounds the number of cases as follows. UPPER BOUND: From date of first death of 12 March, a presumptive case fatality rate between 1% and 4%, and average time to death of 18 days, one might conclude that 25 to 100 persons were infected in Georgia by 23 February. We will consider this to be 128 cases on March 1. However, this case was reported to display relevant comorbidities prior to fatality and died two days after hospitalization, suggesting that a faster than average disease progression. MID: Within the first week of March, seven cases were confirmed in Georgia. We estiamted a case detection rate of 11% in Wuhan when it around the same stage of progression, suggesting 64 cases in Georgia in the first week of March. LOWER BOUND: As of 13 March, 42 cases are known. If these are the majority of cases (i.e. case detection and notification is actually really good) and represent two infection generations, then there were perhaps $\frac{42}{R_0^2} = \frac{42}{2.3^2} \approx 8$ persons infected in Georgia on March 1.

Given widespread awareness about COVID-19 it is assumed that case isolation will be relatively rapid and effective and that the primary impact of interventions is to reduce transmissibility (β). We assume a natural symptomatic infectious period prior to March 11 and fixed at 1.5 days from symptom onset to isolation after that. Interestingly, this isolation rate yields an effective reproduction number right around 1.

We assume average case notificatio take three days due to the lack of testing kits and turnaround time and that all cases are eventually detected.

```r
gamma <- function(z = 11, b=0.143, a0=1/1.5, t){
  # piecewise function
  # default parameters z = 11, b=1/7, a0=1/1.5
  #    z: time at start of intervention (notionally Jan 19, 50 days after notional start on Dec 1 and t
  #    b: intercept (positive)
  #    a0: post intervention isolation ratae
  #    t: time in the model

  gamma <- ifelse(t<=z, gamma <- b, gamma <- a0)
  return(gamma)
}

eta <- function(t) ifelse(t<=55,1/3,1/3)

q <- function(t, w=50, q0=1, q1=1) ifelse(t<=w,q0,q1)

beta <- function(t, w=11, beta0=0.6584, beta.factor=3) ifelse(t<=w,beta0,beta0/beta.factor)
```

```r
evaluate.model <- function(params=list(beta0=0.6584, sigma=1/6.4, z=11, b=0.143, a0=1/1.5, w=40, c=1, d
                                       init = list(S=10520000, E1=0, E2=0, E3=0, E4=0, E5=6, E6=0,
                                                   I1 = 1, I2= 0, I3=0, I4=0, Iu1=0, Iu2=0, Iu3=0, Iu4=
                                                   H=0, Ru=0, C=0),
                           nsims=2, nstep=NULL, start=as.Date("2020-03-01"),today=Sys.Date()){

  if(is.null(nstep)) nstep <- (as.numeric(today-start)+1+28)/params$dt #run simulation from start to cu

  xstart <- c(time=0, unlist(init), cum.time = 0) #initial conditions

  data <- vector(mode='list',length=nsims) #initialize list to store the output

  for (k in 1:nsims) {              #simulate nsims times
    data[[k]] <- as.data.frame(model(xstart,params,nstep))
    data[[k]]$cum.time <- cumsum(data[[k]]$time)

  }

  return(data)
}
```

The function `plot.model` provides automated visualization of model simulations.

```r
plot.model <- function(data, log='y'){

  # process data
  nsims <- length(data)

  for(i in 1:nsims) data[[i]]$I <- data[[i]]$I1 + data[[i]]$I2 + data[[i]]$I3 +
      data[[i]]$I4
  for(i in 1:nsims) data[[i]]$Iu <- data[[i]]$Iu1 + data[[i]]$Iu2 + data[[i]]$Iu3 +
      data[[i]]$Iu4
  for(i in 1:nsims) data[[i]]$E <- data[[i]]$E1 + data[[i]]$E2 + data[[i]]$E3 +
      data[[i]]$E4 + data[[i]]$E5 + data[[i]]$E6

  max.time<-data[[1]]$cum.time[max(which(data[[1]]$I>0))] #maximum time in first simulation
  max.y<-max(data[[1]]$C)       #find max total confirmed cases for plotting range

  # calculate means
  m1 <- m2 <- m3 <- m4 <- m5 <- matrix(nrow=length(data[[1]]$I), ncol=nsims)
  for(i in 1:nsims){
    m1[,i] <- data[[i]]$E
    m2[,i] <- data[[i]]$I+data[[i]]$Iu
   # m3[,i] <- data[[i]]$Iu
    m4[,i] <- data[[i]]$H
    m5[,i] <- data[[i]]$C
  }
  E.mean <- rowMeans(m1)
  I.mean <- rowMeans(m2)
 # Iu.mean <- rowMeans(m3)
  H.mean <- rowMeans(m4)
  C.mean <- rowMeans(m5)

  # colors
  E.col <- rgb(0,1,0,.25)
  I.col <- rgb(1,0,0,.25)
  Iu.col <- rgb(0.5, 0.5, 0, 0.25)
  H.col <- rgb(0,0,1,.25)
  C.col <- rgb(0,0,0,.25)
  E.mean.col <- rgb(0,1,0,1)
  I.mean.col <- rgb(1,0,0,1)
  Iu.mean.col <- rgb(0.5,0.5,0,1)
  H.mean.col <- rgb(0,0,1,1)
  C.mean.col <- rgb(0,0,0,1)

  #set up plot
  plot(I~cum.time,data=data[[1]],xlab='',ylab='Cases',col=1,
       xlim=c(0,max.time),ylim=c(1,max.y), type='n', lty=1, log=log, axes=FALSE) # set up plot

  # add data to plot
  day <- georgia$date - start
  lines(day, cumsum(georgia$cases), type='h', col='lightskyblue', lwd=3, lend='butt' )

  # plot spaghetti
  lines(E~cum.time,data=data[[1]], col=E.col, lty=1)
  lines(I+Iu~cum.time,data=data[[1]], col=I.col, lty=1)
 # lines(Iu~cum.time,data=data[[1]], col=Iu.col, lty=1)
  lines(H~cum.time,data=data[[1]], col=H.col, lty=1)
  lines(C~cum.time,data=data[[1]], col=C.col, lty=1, lwd=1)


  axis(1, at=seq(0,max.time,5), labels=format(start+seq(0,max.time,5), format= '%b %d'))
  axis(2)
  box()

  if(nsims > 1){
  for (k in 2:min(100,nsims)) {              #add multiple epidemics to plot
    lines(E~cum.time, data=data[[k]], col=E.col, type='l', lty=1)
    lines(I+Iu~cum.time, data=data[[k]], col=I.col, type='l', lty=1)
 #   lines(Iu~cum.time, data=data[[k]], col=Iu.col, type='l', lty=1)
    lines(H~cum.time, data=data[[k]], col=H.col, type='l', lty=1)
    lines(C~cum.time, data=data[[k]], col=C.col, type='l', lty=1, lwd=1)
  }

  # plot means
  lines(E.mean~cum.time, data=data[[k]], col=E.mean.col, lty=1)
  lines(I.mean~cum.time, data=data[[k]], col=I.mean.col, lty=1)
#  lines(Iu.mean~cum.time, data=data[[k]], col=Iu.mean.col, lty=1)
  lines(H.mean~cum.time, data=data[[k]], col=H.mean.col, lty=1)
  lines(C.mean~cum.time, data=data[[k]], col=C.mean.col, lty=1)
  }

   legend('topleft', lty=c(1,1,1,1,1), lwd=c(1,1,1,1,1), bty='n', cex=0.75,
         col=c(E.col, I.col, H.col, C.col),
         legend=c('Latent cases', 'Infectious cases', 'Hospitalized',
                  'Cumulative reported cases (Model)'))
}
```

A second function plots just the observed and unobserved cases over time.

```
plot.model2 <- function(data, log='y'){

  # process data
  nsims <- length(data)

  for(i in 1:nsims) data[[i]]$I <- data[[i]]$I1 + data[[i]]$I2 + data[[i]]$I3 +
      data[[i]]$I4
  for(i in 1:nsims) data[[i]]$Iu <- data[[i]]$Iu1 + data[[i]]$Iu2 + data[[i]]$Iu3 +
      data[[i]]$Iu4
  for(i in 1:nsims) data[[i]]$E <- data[[i]]$E1 + data[[i]]$E2 + data[[i]]$E3 +
      data[[i]]$E4 + data[[i]]$E5 + data[[i]]$E6

  max.time<-data[[1]]$cum.time[max(which(data[[1]]$I>0))] #maximum time in first simulation
  #max.y<-max(data[[1]]$Q)       #find max total confirmed cases for plotting range
  max.y <- max(unlist(lapply(data, FUN=function(x) max(x$C))))

  # calculate means
  m1 <- m2 <- m3 <- m4 <- m5 <- matrix(nrow=length(data[[1]]$I), ncol=nsims)
  for(i in 1:nsims){
    m1[,i] <- data[[i]]$E
    m2[,i] <- data[[i]]$I
    m3[,i] <- data[[i]]$Iu
    m4[,i] <- data[[i]]$H
    m5[,i] <- data[[i]]$C
  }

  observed <- m5
  unobserved <- m1 + m2 + m3 + m4

  obs.mean <- rowMeans(observed)
  unobs.mean <- rowMeans(unobserved)
  cum.time <- data[[1]]$cum.time
  # colors
  E.col <- rgb(0,1,0,.25)
  I.col <- rgb(1,0,0,.25)
  Iu.col <- rgb(0.5, 0.5, 0, 0.25)
  H.col <- rgb(0,0,1,.25)
  C.col <- rgb(0,0,0,.25)
  E.mean.col <- rgb(0,1,0,1)
  I.mean.col <- rgb(1,0,0,1)
  Iu.mean.col <- rgb(0.5,0.5,0,1)
  H.mean.col <- rgb(0,0,1,1)
  C.mean.col <- rgb(0,0,0,1)

  #set up plot
  plot(obs.mean~cum.time, xlab='',ylab='Cases',col=1,
       xlim=c(0,max.time),ylim=c(1,max.y), type='n', lty=1, log=log, axes=FALSE) # set up plot

  # add data to plot
  day <- georgia$date - start
  lines(day, cumsum(georgia$cases), type='h', col='lightskyblue', lwd=3, lend='butt' )

  # plot spaghetti
  lines(unobserved[,1]~cum.time, col=E.col, lty=1)
  lines(observed[,1]~cum.time, col=C.col, lty=1, lwd=1)

  axis(1, at=seq(0,max.time,5), labels=format(start+seq(0,max.time,5), format= '%b %d'))
  axis(2)
  box()

  if(nsims > 1){
  for (k in 2:min(100,nsims)) {              #add multiple epidemics to plot
    lines(unobserved[,k]~cum.time,  col=E.col, type='l', lty=1)
    lines(observed[,k]~cum.time, col=C.col, type='l', lty=1, lwd=1)
  }

  # plot means
  lines(unobs.mean~cum.time, col=E.mean.col, lty=1)
  lines(obs.mean~cum.time, col=C.mean.col, lty=1)
  }

   legend('topleft', lty=c(1,1), lwd=c(1,1), bty='n', cex=0.75,
         col=c(E.col, C.col),
         legend=c('Unobserved cases', 'Cumulative reported cases (Model)'))
}
```

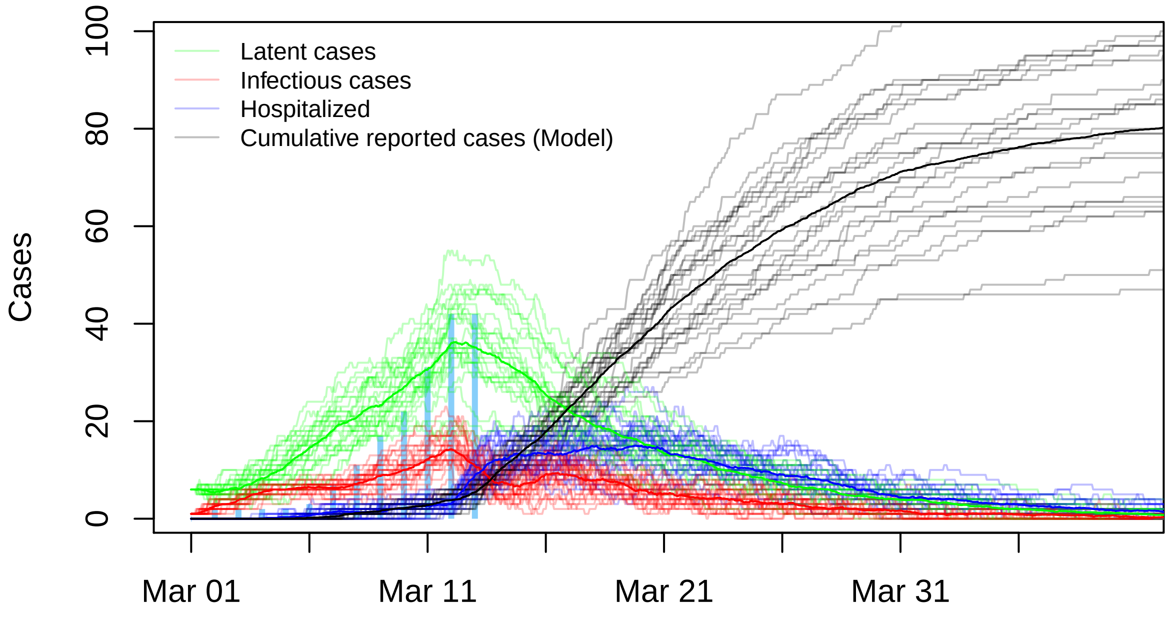
A third plotting function summarizes the model-based distribution of cases and notifications at the current time.

```
plot.summary <- function(data){
   # process data
  nsims <- length(data)

  for(i in 1:nsims) data[[i]]$I <- data[[i]]$I1 + data[[i]]$I2 + data[[i]]$I3 +
      data[[i]]$I4
  for(i in 1:nsims) data[[i]]$Iu <- data[[i]]$Iu1 + data[[i]]$Iu2 + data[[i]]$Iu3 +
```

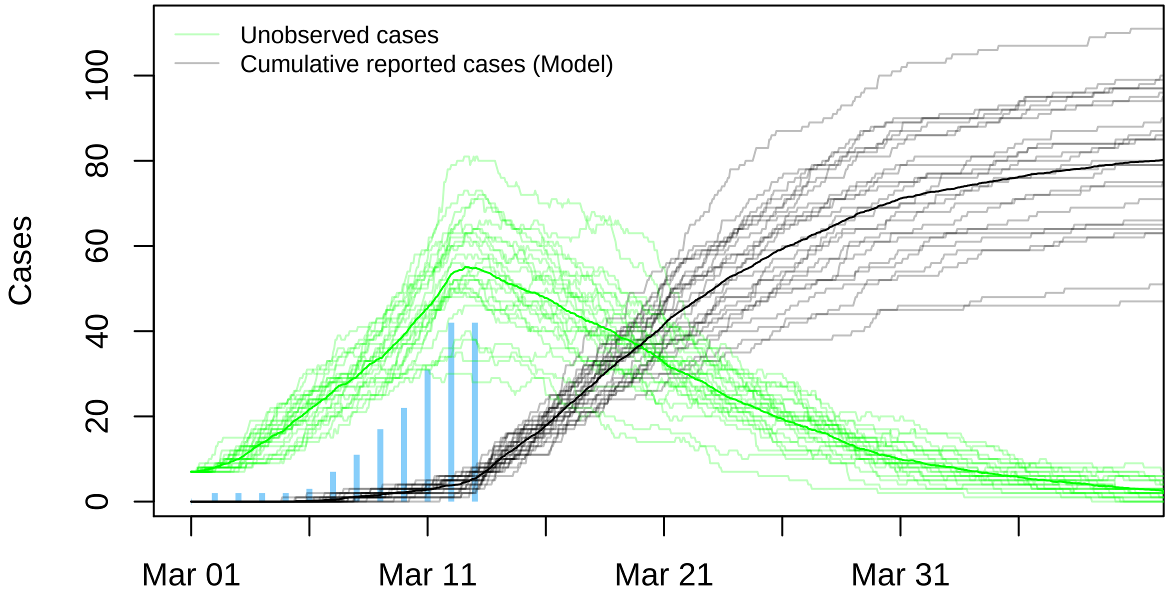
```r
      data[[i]]$Iu4
  for(i in 1:nsims) data[[i]]$E <- data[[i]]$E1 + data[[i]]$E2 + data[[i]]$E3 +
      data[[i]]$E4 + data[[i]]$E5 + data[[i]]$E6

  max.time<-data[[1]]$cum.time[max(which(data[[1]]$I>0))] #maximum time in first simulation
  max.y<-max(data[[1]]$I)       #find max total confirmed cases for plotting range

  # calculate means
  m1 <- m2 <- m3 <- m4 <- m5 <- matrix(nrow=length(data[[1]]$I), ncol=nsims)
  for(i in 1:nsims){
    m1[,i] <- data[[i]]$E
    m2[,i] <- data[[i]]$I
    m3[,i] <- data[[i]]$Iu
    m4[,i] <- data[[i]]$H
    m5[,i] <- data[[i]]$C
  }
  E.mean <- rowMeans(m1)
  I.mean <- rowMeans(m2)
  Iu.mean <- rowMeans(m3)
  H.mean <- rowMeans(m4)
  C.mean <- rowMeans(m5)

  # colors
  E.col <- rgb(0,1,0,.25)
  I.col <- rgb(1,0,0,.25)
  Iu.col <- rgb(0.5, 0.5, 0, 0.25)
  H.col <- rgb(0,0,1,.25)
  C.col <- rgb(0,0,0,.25)
  E.mean.col <- rgb(0,1,0,1)
  I.mean.col <- rgb(1,0,0,1)
  Iu.mean.col <- rgb(0.5,0.5,0,1)
  H.mean.col <- rgb(0,0,1,1)
  C.mean.col <- rgb(0,0,0,1)

  #set up plot
  j <- max(which(data[[1]]$time < today.day))

  h1 <- hist(log10(m1[j,]+m2[j,]+m3[j,]+m4[j,]), plot=FALSE)  # total unnotifed cases in the community
  h2 <- hist(log10(m5[j,]), plot=FALSE)         # total case notifications

  xlim <- c(floor(min(c(h1$mids, h2$mids))),ceiling(max(c(h1$mids, h2$mids))))
  ylim <- c(0, ceiling(max(c(h1$density, h2$density))))

  plot(xlim, ylim, type='n', axes=FALSE, xlab='Outbreak size', ylab='Probability density', main=paste('
  lines(h1$mids, h1$density, type='l', lwd=3, lend='butt', col=rgb(0.5,0.5,0, 0.5))
  lines(h2$mids, h2$density, type='l', lwd=3, lend='butt', col=C.col)
  legend('topleft', lty=1, col = c(rgb(0.5,0.5,0, 0.5), C.col), legend=c('Unobserved cases', 'Case noti
  axis(1, seq(min(xlim), max(xlim), by=1), labels=formatC(10^seq(min(xlim), max(xlim), by=1), format='d
  box()
}
```

Here we simulate for Georgia, starting with one initial case on March 1, 2020.

```
start <- as.Date('03/01/2020',format='%m/%d/%Y')
set.seed(1292020)                    #set seed
out1 <- evaluate.model(params=list(beta0=0.6584, sigma=1/6.4, z=11, b=0.143, a0=1/1.5, w=40, c=1, dt=0.
                                  init = list(S=10520000, E1=0, E2=0, E3=0, E4=0, E5=6, E6=0,
                                              I1 = 1, I2= 0, I3=0, I4=0, Iu1=0, Iu2=0, Iu3=0, Iu4=
                                              H=0, Ru=0, C=0),
                       nsims=25, nstep=NULL, start=as.Date("2020-03-01"))


plot.model(out1, log='')
```



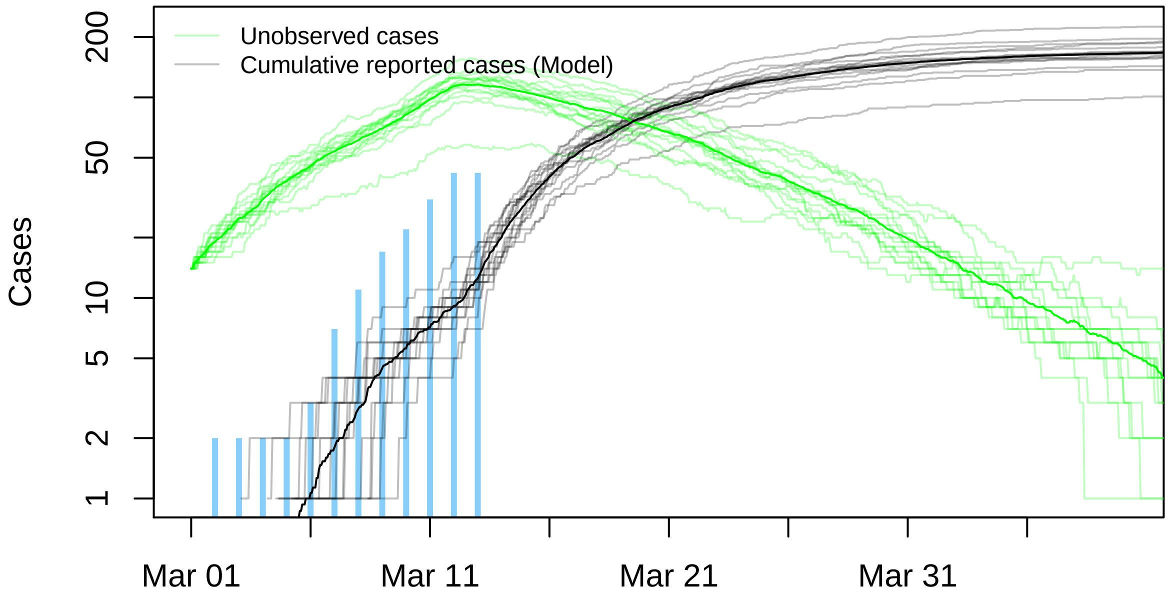
```
plot.model2(out1, log='')
```



Now, we look at our more realistic scenarios of $I_1 = 8$, $I_1 = 64$, and $I_1 = 128$. It would appear that the total case reports of 42 by March 13 are most consistent with the intermediate scenario.

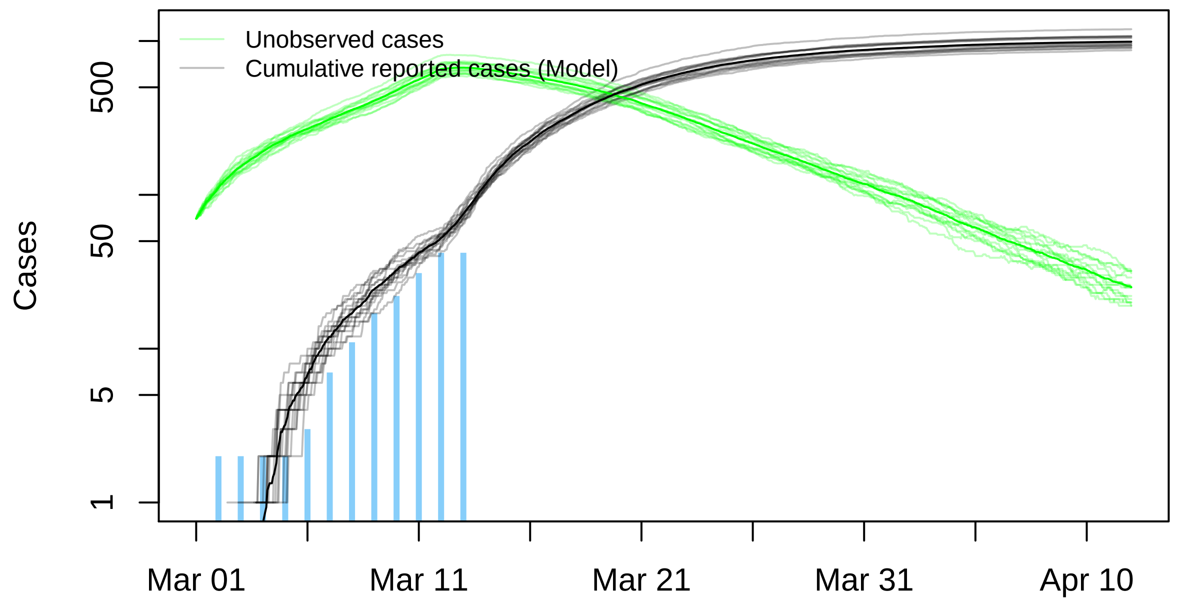
```
set.seed(1292020)                      #set seed
out8 <- evaluate.model(params=list(beta0=0.6584, sigma=1/6.4, z=11, b=0.143, a0=1/1.5, w=40, c=1, dt=0.
                                       init = list(S=10520000, E1=0, E2=0, E3=0, E4=0, E5=6, E6=0,
                                                   I1 = 8, I2= 0, I3=0, I4=0, Iu1=0, Iu2=0, Iu3=0, Iu4=
                                                   H=0, Ru=0, C=0),
                           nsims=15, nstep=NULL, start=as.Date("2020-03-01"))
plot.model2(out8, log='y')
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 42 y values <= 0 omitted
## from logarithmic plot
```



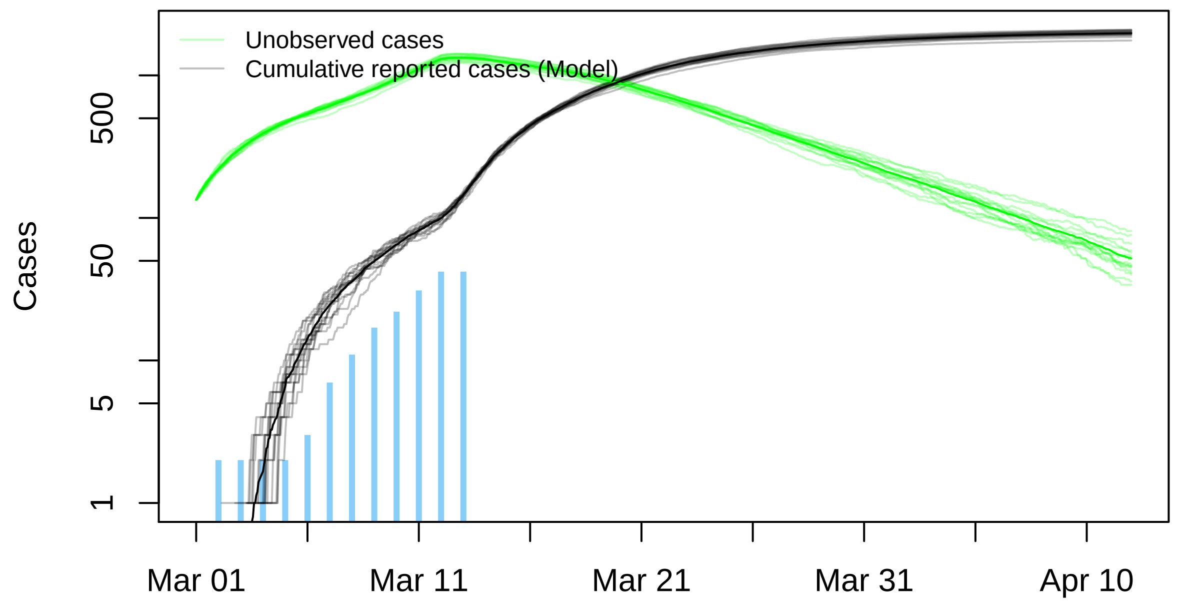
```
out64 <- evaluate.model(params=list(beta0=0.6584, sigma=1/6.4, z=11, b=0.143, a0=1/1.5, w=40, c=1, dt=0
                                       init = list(S=10520000, E1=0, E2=0, E3=0, E4=0, E5=6, E6=0,
                                                   I1 = 64, I2= 0, I3=0, I4=0, Iu1=0, Iu2=0, Iu3=0, Iu4=
                                                   H=0, Ru=0, C=0),
                           nsims=15, nstep=NULL, start=as.Date("2020-03-01"))
plot.model2(out64, log='y')
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 28 y values <= 0 omitted
## from logarithmic plot
```



```
out128 <- evaluate.model(params=list(beta0=0.6584, sigma=1/6.4, z=11, b=0.143, a0=1/1.5, w=40, c=1, dt=
                                       init = list(S=10520000, E1=0, E2=0, E3=0, E4=0, E5=6, E6=0,
                                                   I1 = 128, I2= 0, I3=0, I4=0, Iu1=0, Iu2=0, Iu3=0, Iu
                                                   H=0, Ru=0, C=0),
                           nsims=15, nstep=NULL, start=as.Date("2020-03-01"))
plot.model2(out128, log='y')
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 23 y values <= 0 omitted
## from logarithmic plot
```


Unobserved cases
Cumulative reported cases (Model)
Cases
Mar 01
Mar 11
Mar 21
Mar 31
Apr 10