Applying SIR model in Covid-19 data

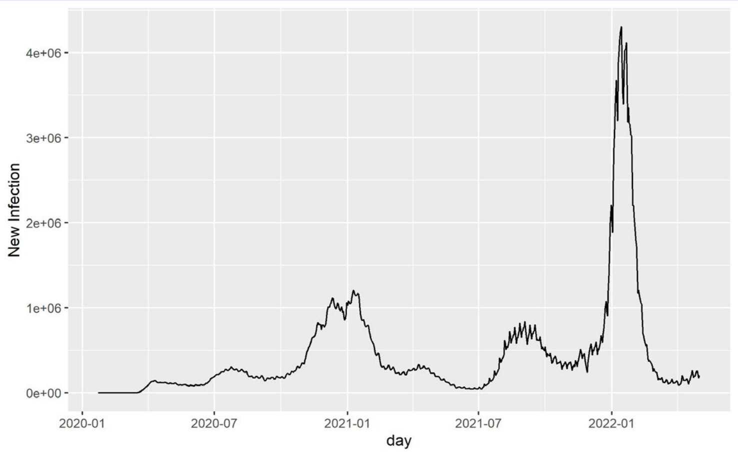
Youngwoo Kwon

Goal

- Analyze Covid-19 in the United States
- Find models that can represent the Covid-19 data

Data

- Covid-19 in the United States from 2020 to 2022
- Our World in Data COVID-19 repository



1. SIR model with Bayesian Method

Continuous-time SIR model

- The SIR model

$$N = S(t) + I(t) + R(t)$$
$$\frac{dN}{dt} = \frac{dS(t)}{dt} + \frac{dI(t)}{dt} + \frac{dR(t)}{dt}$$

- Transition Probability

$$p(\Delta t) = \begin{cases} \frac{\beta s i}{N}\Delta t, & \text{if (s, i)} \xrightarrow{\Delta t} (s-1, i+1) \\ \gamma i \Delta t, & \text{if (s, i)} \xrightarrow{\Delta t} (s, i-1) \\ 1 - \left[\frac{\beta s i}{N}\Delta t + \gamma i \Delta t\right], & \text{if (s, i)} \xrightarrow{\Delta t} (s, i) \end{cases}$$

Discrete-time approximation

* To deal with real-world data, we need some discrete-time approximation

$$i_d \sim \text{Poisson}\left(\frac{\beta\ S(t_d)\ I(t_d)}{N}(t_d - t_{d-1})\right)$$

$$r_d \sim \text{Poisson}\left(\gamma i(t_d - t_{d-1})\right)$$

Bayesian Framework

- What we want to find is $p(\beta|D)$, $p(\gamma|D)$
- Bayesian Framework

$$p(\beta \mid \mathcal{D}) \propto_\beta \prod_{d \in D} p_{e_i}(i_d|i_{d-1}, \phi)\ p(\beta)$$

$$p(\beta \mid \mathcal{D}) \propto_\beta \left[\prod_{d \in D} \frac{\left(\frac{\beta S(t_d) I(t_d) \Delta t}{N} \right)^{i_d} \ \exp\left\{ -\frac{\beta S(t_d) I(t_d) \Delta t}{N} \right\}}{i_d} \right] p(\beta)$$

$$p(\gamma \mid \mathcal{D}) \propto_\gamma \prod_{d \in D} p_{e_i}(i_r|i_{d-1}, \phi)\ p(\gamma)$$

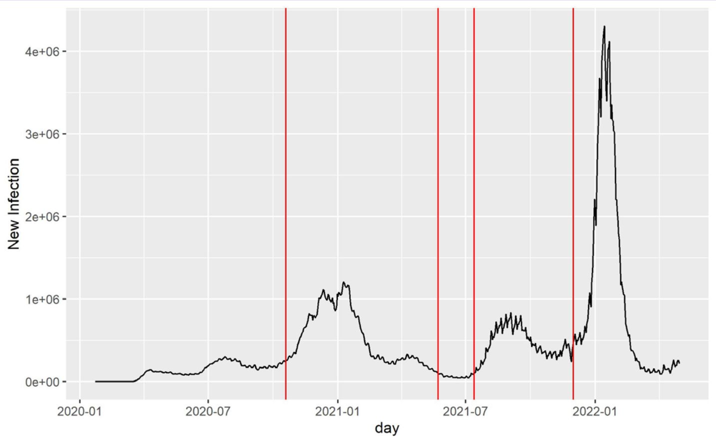
$$p(\gamma \mid \mathcal{D}) \propto_\gamma \left[\prod_{d \in D} \frac{(\gamma i \Delta t)^{i_r} \ \exp\{-\gamma i \Delta t\}}{i_r} \right] p(\gamma)$$

Metropolis-Hasting

- Markov chain Monte Carlo (MCMC) method for obtaining a sequence of random samples from a probability distribution.
- For $\phi = \beta, \gamma$
  a. Start with ϕ_0
  b. For $i = 1, 2, \dots, n$
     i. Draw ϕ^* from $q(\phi_{i-1})$
     ii. Compute $a = \frac{L(D,\phi^*)q(\phi_{i-1}|\phi^*)}{L(D,\phi_{i-1})q(\phi^*|\phi_{i-1})}$
     iii. If $a > 1$, accept ϕ^*
     iv. If $0 < a < 1$, accept ϕ^* with probability a.

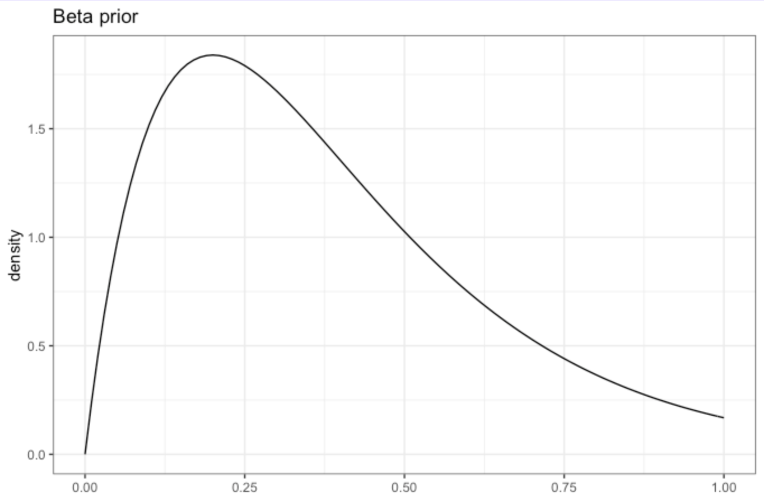
Apply to the real data

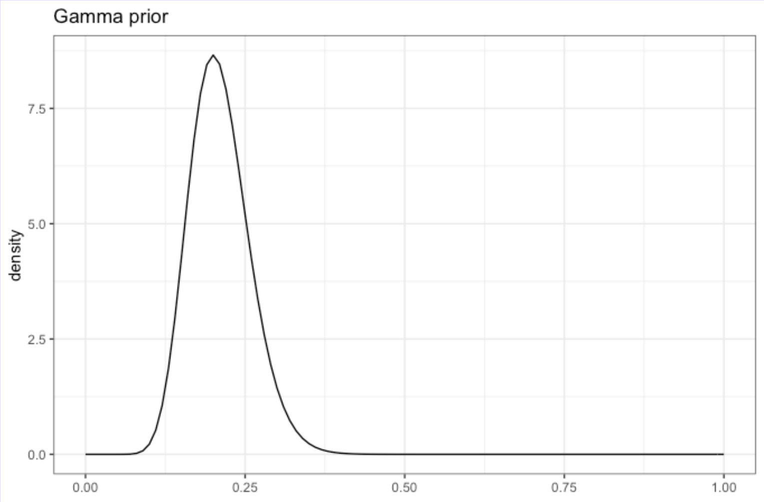
- Change of β over pandemic
  a. Mask restriction in US (2.5 -> 3) 2020.10.19
  b. Fully vaccinated rate reached 50% 2021.01.13
  c. Mask restriction in US (3 -> 2.5) 2021.05.22
  d. Omicron in US 2021.12.01
- Constant γ over pandemic



Prior distribution selection

- Gamma Distribution for β and γ





Result

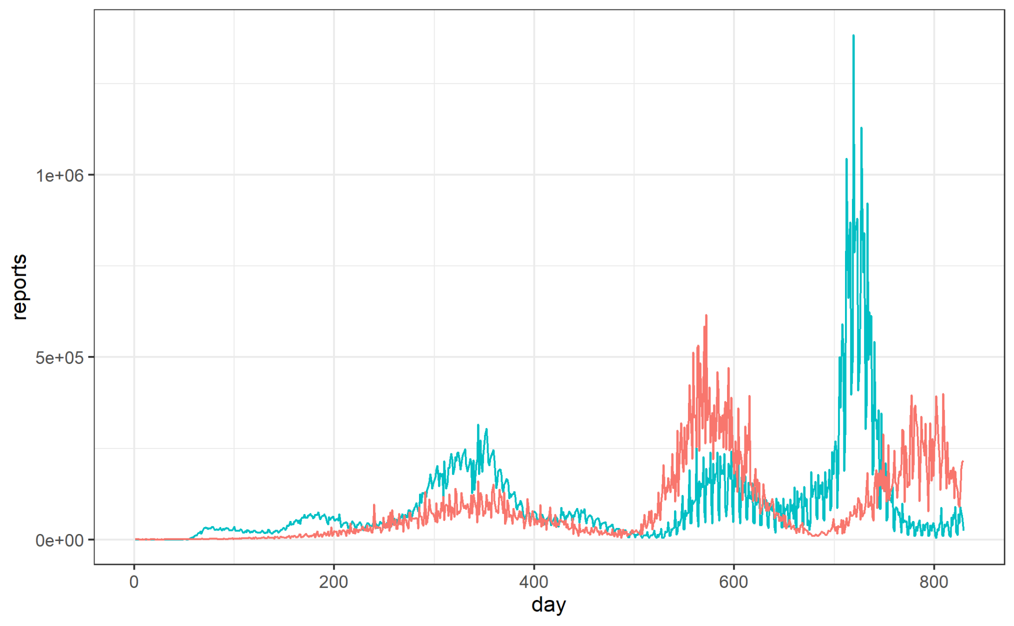
- Change of β over time

Table 1: Posterior means

Parameter	Posterior mean
β_1	0.234
β_2	0.229
β_3	0.269
β_4	0.251
β_5	0.253
γ	0.212

Result

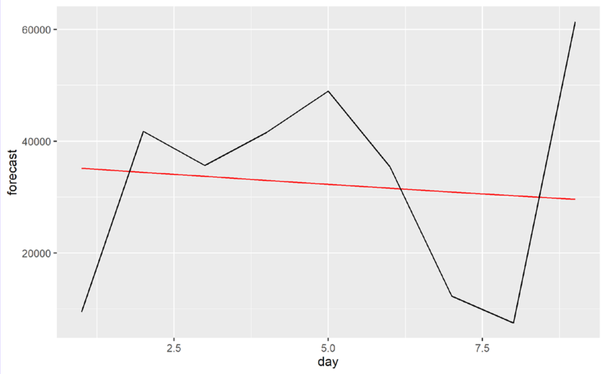
- Blue: real data
- Red: simulation data



Forecasting Covid cases

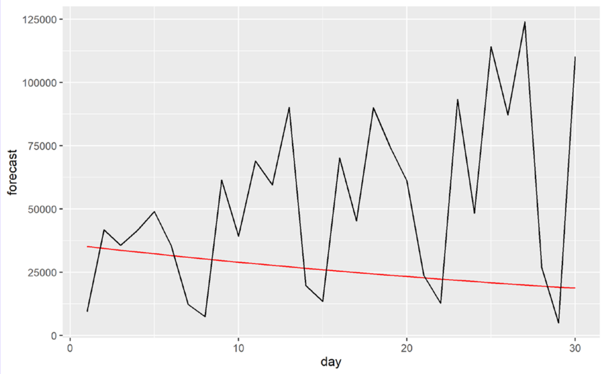
- 9 days prediction

pred	32346.44
actual	32680.78



- 30 days prediction

pred	26161.97
actual	52383.63



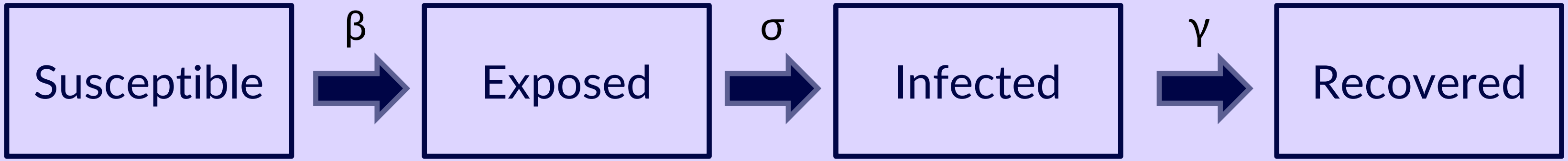
Summary

- Divided data into 5 parts
- Short computation time
- Difference between the model and actual infection
    - Change of $\beta/\gamma/N$ over time
    - Gap between the actual infection and the S-I-R process
    - A person who is infected once cannot be infected again
    - Need a more complex model

2. SEIREIR model

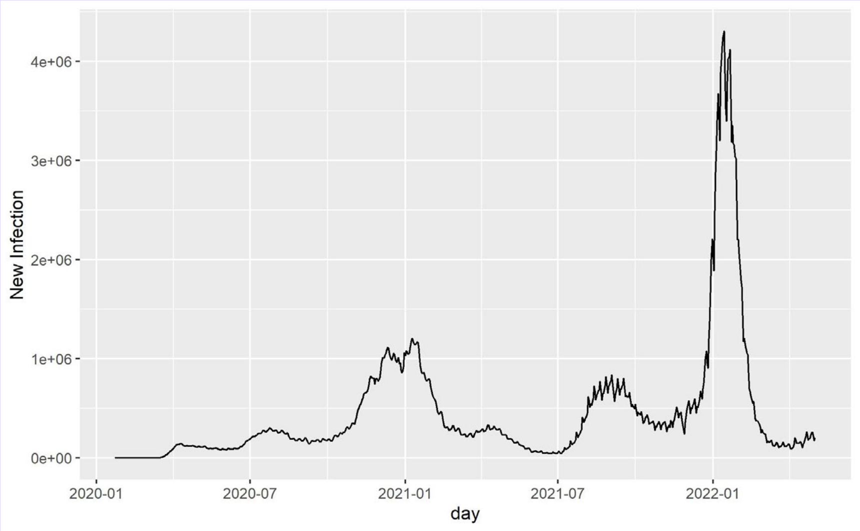
SEIR model

- Significant latency period during which individuals have been infected but not yet infectious.



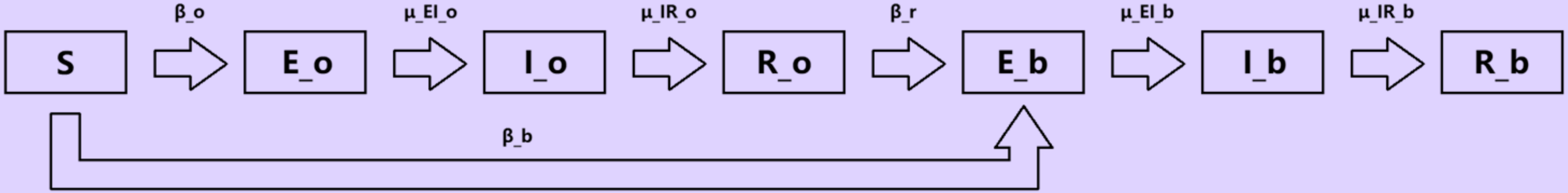
SEIREIR model

- Two covid19 pandemic seasons
    - before omicron (2021-01)
    - after omicron (2022-02)



SEIREIR model

- Two EIR processes in the model
  - E_o, I_o, R_o : States for Covid19 before omicron
  - E_b, I_b, R_b : States for Covid19 after omicron



SEIR model

1) Build a model

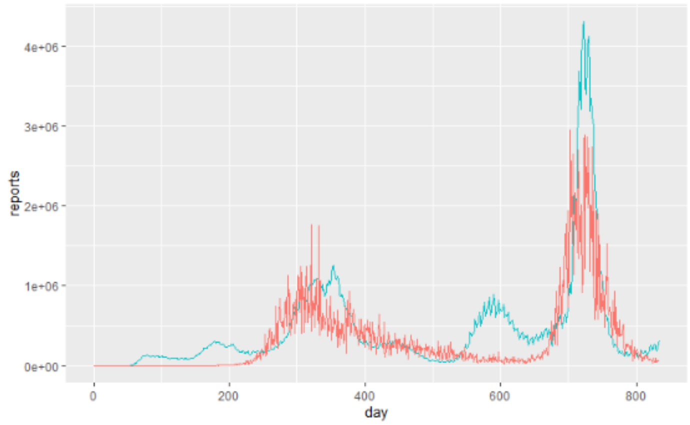
```
data %>%
  dplyr::select(day,reports=cases) %>%
  filter(day<=834) %>%
  pomp(
    times="day",t0=0,
    rprocess=euler(seireir_step,delta.t=1),
    rinit=seireir_init,
    rmeasure=rmeas,
    dmeasure=dmeas,
    accumvars="H",
    statenames=c("S","E_o","I_o","R_o","E_b","I_b","R_b","H"),
    paramnames=c("Beta_o","Beta_b","Beta_r","mu_EI_o","mu_IR_o",
                 "Beta_or","mu_IR_r",
                 "mu_EI_b","mu_IR_b","eta","rho","k","N"),
    params=c(Beta_o=0.2, Beta_b = 14,Beta_or = 0.22, Beta_r = 0.5, mu_EI_o= 0.66,
             mu_IR_o = 0.185,mu_EI_b = 0.2,mu_IR_r = 0.0085,
             mu_IR_b=0.04257,rho=0.996,k=9,eta=0.4,N=332915074)
  ) -> measSEIREIR
```

SEIR model

2) Simulate Graphs

````
```{r echo=TRUE}
measSEIREIR %>%
simulate(
params=c(Beta_o=0.2, Beta_b = 14,Beta_or = 0.22, Beta_r = 0.5, mu_EI_o= 0.66,
            mu_IR_o = 0.185,mu_EI_b = 0.2,mu_IR_r = 0.0085,
            mu_IR_b=0.04257,rho=0.996,k=9,eta=0.4,N=332915074),
nsim=1,format="data.frame",include.data=TRUE
) -> sims

sims %>%
ggplot(aes(x=day,y=reports,group=.id,color=.id=="data"))+
geom_line()+
guides(color="none")
```
````



SEIR model

3) Local search

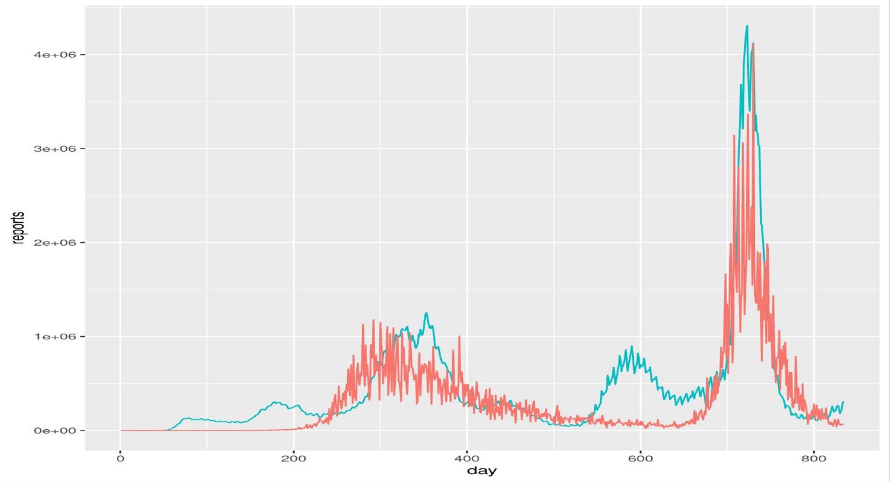
```
```{r echo=TRUE}
params <- c(Beta_o=0.2, Beta_b = 14,Beta_or = 0.22, Beta_r = 0.5, mu_EI_o= 0.66,
            mu_IR_o = 0.185,mu_EI_b = 0.2,mu_IR_r = 0.0085,
            mu_IR_b=0.04257,rho=0.996,k=9,eta=0.4,N=332915074)
measSEIREIR %>%
  pomp(
    paramnames=c("Beta_o","Beta_b","Beta_r","Beta_or","rho","mu_EI_o","mu_EI_b","eta"),
  ) -> measSEIREIR2

bake(file="local_search.rds",{
  registerDoRNG(482947940)
  foreach(i=1:5,.combine=c) %do% {
    library(pomp)
    library(tidyverse)
    measSEIREIR2 %>%
      mif2(
        params=params,
        Np=1000, Nmif=50,
        cooling.fraction.50=0.5,
        rw.sd=rw.sd(Beta_o=0.0001,Beta_b = 0.001,Beta_or = 0.0001,
                    Beta_r = 0.0001,mu_EI_o=0.0001, mu_IR_o = 0.00001, mu_EI_b = 0.00001,
             rho=0.0001,eta=ivp(0.0001),mu_IR_r = 0.000001,
             mu_IR_b=0.000001)
      )
  } -> mifs_local
}) -> mifs_local
```
```

SEIR model

4) Global Search

```
```{r echo=TRUE}
set.seed(2062379496)
runif_design(
  lower = c(Beta_o=1,Beta_b = 50,Beta_or = 0.5,
            Beta_r = 30,mu_EI_o=0.1,mu_EI_b = 0.03,
            rho=0.2,eta = 0),
  upper = c(Beta_o=100,Beta_b = 140,Beta_or = 10,
            Beta_r = 70,mu_EI_o=0.3,mu_EI_b = 0.2,
            rho=1,eta = 1),
nseq=160
) -> guesses
```
```

```
```{r include=FALSE, eval=FALSE}
bake(file="global_search.rds",{
  foreach(guess=iter(guesses,"row"), .combine=rbind) %dopar% {
  library(pomp)
  library(tidyverse)
  mf1 %>%
    mif2(params=c(guess,fixed_params)) %>%
    mif2(Nmif=50) -> mf
  replicate(
    10,
    mf %>% pfilter(Np=1000) %>% logLik()
    ) %>%
    logmeanexp(se=TRUE) -> ll
  mf %>% coef() %>% bind_rows() %>%
    bind_cols(loglik=ll[1],loglik.se=ll[2])
} -> results
  results
}) %>%
  filter((is.finite(loglik)))-> global_results


save(list = c('global_results'),file = 'global.RData')
```
```

SEIREIR model

- Local Search

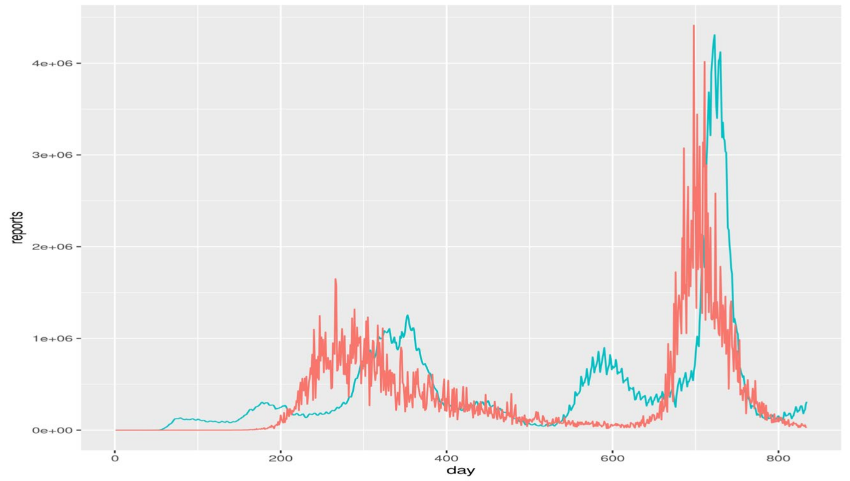
coef.	β_o	β_b	β_r	$\mu_{EI,o}$	$\mu_{IR,o}$	$\mu_{EI,b}$
value	0.198	13.23	0.499	0.663	0.184	0.200
coef.	$\mu_{IR,r}$	$\mu_{IR,b}$	ρ	k	η	N
value	0.009	0.042	0.996	9	0.400	3.3e8



SEIREIR model

- Global Search

coef.	β_o	β_b	β_r	$\mu_{EI,o}$	$\mu_{IR,o}$	$\mu_{EI,b}$
value	0.213	12.62	0.471	0.652	0.171	0.204
coef.	$\mu_{IR,r}$	$\mu_{IR,b}$	ρ	k	η	N
value	0.009	0.040	0.996	9	0.446	3.3e8



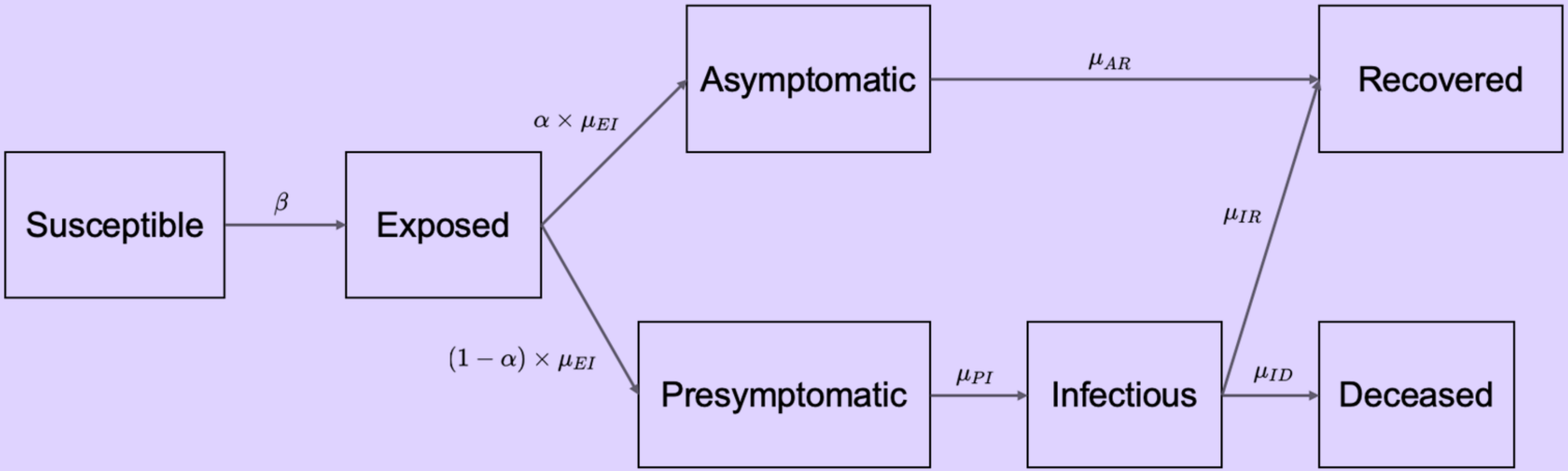
Summary

- Model and real data tend to match
- Long computation time
- There is still a gap between the SEIR model and the actual covid19 infection

3. SEAPIRD model

SEAPIRD model

- More focused on after omicron cases, with a complicated framework



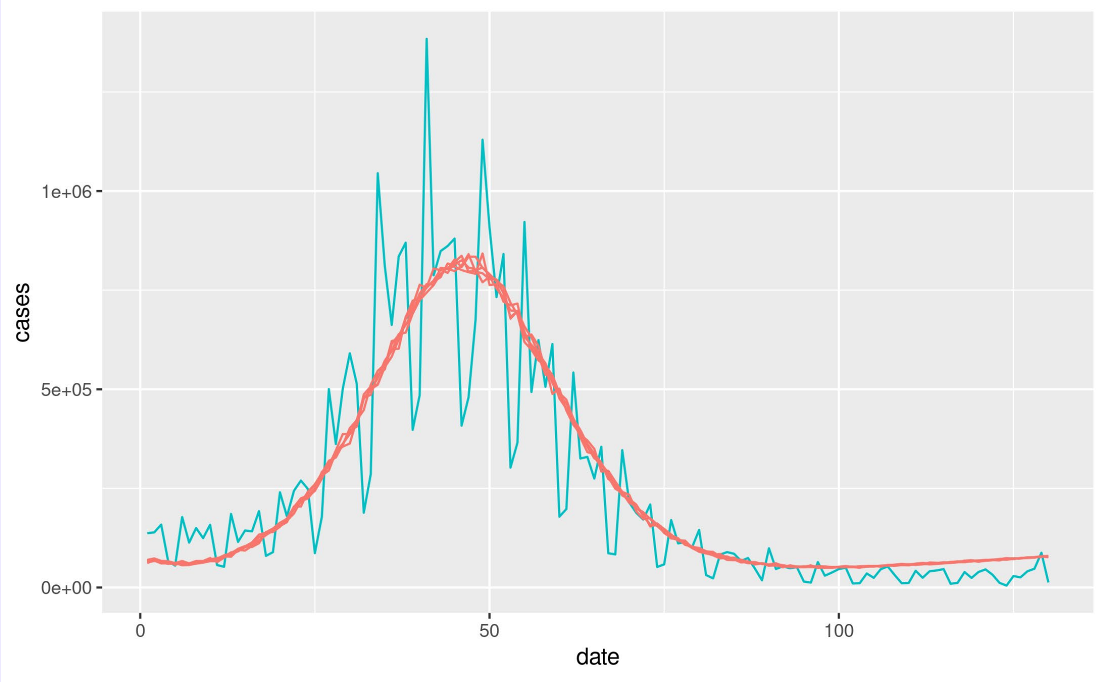
SEAPIRD model

- Local search

coef.	β	μ_{IR}	μ_{ID}	μ_{EI}	α
value	0.1515	8.000e-5	5.000e-6	0.8500	0.0866
coef.	μ_{AR}	μ_{PI}	ρ	N	τ
value	0.1492	0.0140	0.9961	3.33e8	1000

SEAPIRD model

- Local search



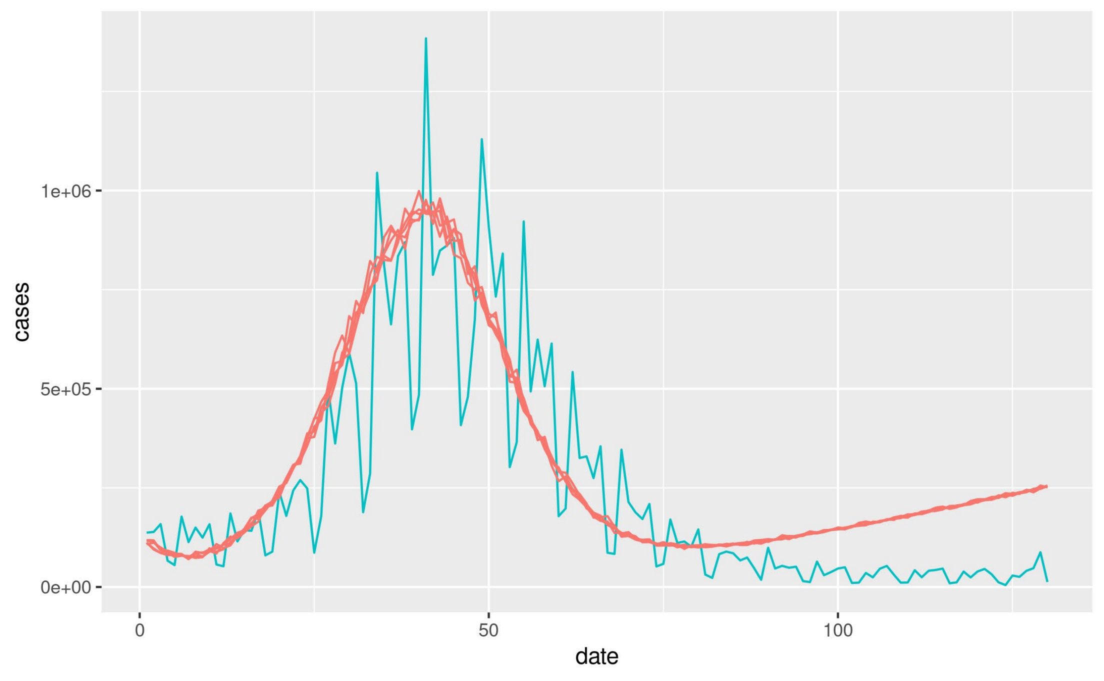
SEAPIRD model

- Global search

coef.	β	μ_{IR}	μ_{ID}	μ_{EI}	α
value	0.1619	1.784e-4	1.672e-6	1.5506	0.0869
coef.	μ_{AR}	μ_{PI}	ρ	N	τ
value	0.2562	0.0139	0.9990	3.33e8	1000

SEAPIRD model

- Global search



Summary

- SEAPIRD model is more similar to the real data
- Short computation time & more accurate results with a shorter time interval

4. Conclusion

Conclusion

- Applied models to real COVID19 data in the U.S.
  - SIR model with Bayesian method
  - SEIREIR model
  - SEAPIRD model
- Considered the process of the model, computation time, and similarity between the simulation and the real world.
- Continuous changes of β, N, γ are not reflected in the models
- Coefficients in the models are lagging indicators. There will be a gap between the prediction and the actual covid19 progress.