CEACOV RSA - Process Output Files

August 10, 2020

1 Overview

This R notebook takes CEACOV output, and performs the operations we are handling "outside the model" (influenza-like illness, contact trancing, and costs).

[58]:
```
library(ggplot2)
```

[59]:
```
# Read in output files from user-specified batch of runs

mypath <- getwd()
batchname <- "R28 8.7.20"
myfolders <- c("BC",
               "CT",
               "CT+IC",
               "CT+IC+MSS",
               "CT+IC+QC",
               "CT+IC+QC+MSS")

fullpaths <- paste(mypath,batchname,sep="/")
fullpaths <- paste(fullpaths,myfolders,sep="/")
fnames <- paste(fullpaths,myfolders,sep="/")

out1 <- paste(fnames,".tsv",sep="")
out2 <- paste(fnames,"_state_data.tsv",sep="")

# Column headings associated with output from model version v0.6_9_intvs
h1 <- "v6_9_intvs_headings.csv"
h2 <- "v6_9_intvs_state_data_headings.csv"
```

[61]:
```
# This function combines the two output files generated by CEACOV into a single␣
↪dataframe

read_ceacov_output<- function(f1,f2,h1,h2){

    df1 <- read.table(f1,sep="\t",header=FALSE)
    df1names <- read.csv(h1)
    colnames(df1) <- colnames(df1names)
    df2 <- read.table(f2,sep="\t",header=FALSE)
```

```
    df2names <- read.csv(h2)
    colnames(df2) <- colnames(df2names)
    df <- cbind(df1,df2)

    return(df)
}
```

[62]:
```
df1 <- read_ceacov_output(out1[1],out2[1],h1,h2) # BC
df2 <- read_ceacov_output(out1[2],out2[2],h1,h2) # CT
df3 <- read_ceacov_output(out1[3],out2[3],h1,h2) # CT+IC
df4 <- read_ceacov_output(out1[4],out2[4],h1,h2) # CT+IC+MSS
df5 <- read_ceacov_output(out1[5],out2[5],h1,h2) # CT+IC+QC
df6 <- read_ceacov_output(out1[6],out2[6],h1,h2) # CT+IC+QC+MSS
```

2 Influenza-like illness and mass symtom screening

Susceptible or recovered patients can present to care with influenza-like illness, which we handle outside of the model. Using age-statified data from AHRI (see table), we are able to determine the probability that under normal circumstances (i.e., when there's no pandemic) about 1.16% of people in KwaZulu-Natal are expected to display symtoms of ILI on a given day ($P(\text{ILI}) = 0.0116$).

Age Group	0-19	20-59	60+
Prevalence of ILI	0.0036	0.0135	0.0445
Fraction of Population in KZN	0.47	0.44	0.09

We assume that the average duration of ILI symptoms is 5 days, that everyone has mild/moderate symptoms, and that a someone with ILI has a 30% chance of presenting to care due to symptoms over the duration of their health state. This translates to a daily probability of presenting to care due to symptoms of

$$P(\text{SYM}) = 1 - (1 - 0.3)^{1/5} = 6.89\%$$

It is also important to account for the effect that ILI has on the specificity of mass symptom screening for COVID-19. Anyone with ILI will be considered a potential case of SARS-CoV-2 by a mass symptom screen, which results in them presenting to care for a PCR test (which will be negative).

If we assume that the entire population of KwaZulu-Natal can be symtom screened twice a year, the daily probability of being symtom screened is given by

$$P(\text{MSS}) = \frac{100\%}{(365 \text{ days per year} \,/\, 12 \text{ months per year})\,(6 \text{ months})} = 0.548\%$$

I assume that presenting to care due to ILI symtoms or due to a symtom screening event are independent but non-mutually exclusive events. For intervention strategies that feature MSS, the probability of someone who is susceptible or recovered presenting to care with ILI is given by.

$$P(\text{present to care with ILI}) = P(\text{ILI}) \times [P(SYM) + P(MSS) - P(SYM)P(MSS)]$$

[63]:
```
age_dist <- c(0.47,0.44,0.09) # Age distribution
KZN_pop <- 11531628

p_MSS <- 1/(365/12*6) # Daily prob of MSS
p_ILI_A1 <- 0.0036    # Prevalence of ILI 0-19
p_ILI_A2 <- 0.0135    # Prevalence of ILI 20-59
p_ILI_A3 <- 0.0445    # Prevalence of ILI 60+

p_ILI_by_age <- c(p_ILI_A1,p_ILI_A2,p_ILI_A3)
p_ILI_overall <- sum(age_dist*p_ILI_by_age)

t_hs <- 5                              # ILI duration
p_pc_hs <- 0.3                         # Probability of pc over ILI duration
p_pc_sym <- 1 - (1 - p_pc_hs)^(1/t_hs) # Daily prob of pc due to symptoms

# No MSS
p_pc_ILI_no_MSS <- p_pc_sym*p_ILI_overall

# MSS
p_pc_ILI_MSS <- (p_pc_sym + p_MSS - p_pc_sym*p_MSS)*p_ILI_overall
```

3 Contact Tracing

Another way in which non-infected individuals can present to care is through contact tracing. The probability of presenting to care in the manner is linked to the number of positive PCR tests on a given day in our model—the event that initiates a contact trace. A number that is analagous to the basic reproduction number is the expected number of negative contacts traced per positive PCR test, which we'll call η_0.

The July 26th COVID-19 KwaZulu-Natal situation report contained the following data related to contact tracing:

Description	Value
Total cases	64061
Contacts identified, traced, and tested	50757
Number of contacts testing positive	2152
Number of contacts testing negative	48605

A PCR test has four possible outcomes: a true positive result, a false positive result, true negative result, or false negative result. The total number of positive and negative tests recorded are given by

$$\text{Number of Positive Tests} = \text{TP} + \text{FP} = 2152$$

$$\text{Number of Negative Tests} = \text{FN} + \text{TN} = 48605$$

In order to solve the above equations, we need to know the sensitivity (TPR) and specificity (TNR) of a PCR test, which are defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Substituting in the sensitivity and specificity of our test gives us a system of linear equations, shown below, which we can use to solve for the number of true positives, true negatives, false positives, and false negatives in our dataset.

$$\text{TP} + \left(\frac{1 - \text{TNR}}{\text{TNR}}\right) \text{TN} = 2152$$

$$\left(\frac{1 - \text{TPR}}{\text{TPR}}\right) \text{TP} + \text{TN} = 48605$$

[64]:
```
contacts_test_pos = 2152   # Number of positive PCR tests among contacts traced
contacts_test_neg = 48605  # Number of negative PCR tests among contacts traced
TPR = 0.7                  # PCR test sensitivity
TNR = 1.0                  # PCR test specificity

# Set up and solve a system of linear equations

A = matrix(nrow=2,ncol=2)
A[1,1] = 1
A[1,2] = (1-TNR)/TNR
A[2,1] = (1-TPR)/TPR
A[2,2] = 1

b = matrix(nrow=2,ncol=1)
b[1] = contacts_test_pos
b[2] = contacts_test_neg

x = solve(A,b)

TP = x[1]
TN = x[2]
FP = contacts_test_pos - TP
FN = contacts_test_neg - TN

contacts_with_covid = TP + FN      # Number of contacts who have covid
contacts_without_covid = TN + FP   # Number of contacts who do not have covid
```

After solving for the number of true positives, true negatives, false positives, and false negatives in our dataset, we can determine the number of contacts traced who have / do not have SARS-CoV-2 as

$$\text{Number of infected contacts traced} = \text{TP} + \text{FN} \approx 3074$$

$$\text{Number of non-infected contacts traced} = \text{TN} + \text{FP} \approx 47683$$

We can now use this information to come up with a rough estimate of η_0, the expected number of non-infected contacts traced per positive PCR result. To do this, we'll divide the number of non-infected contacts traced by the number of confirmed cases of COVID-19 that aren't due to contact tracing (the "original" cases in our dataset).

$$\eta_0 = \frac{47683}{64061 - 2152} \approx 0.77$$

Much like the reproduction number, the number of non-infected contacts traced per positive PCR result is expected to evolve over the course of the pandemic, and should reflect the prevalence of active disease within the population. I handle this by modifying η_0 to make it a function of time

$$\eta(t) = \eta_0 \cdot \frac{S(t) + R(t)}{N_0}$$

where $S(t)$ and $R(t)$ represent the number of susceptible and recovered individuals at a given timepoint in the simulation, and N_0 is the total number of people alive in our system at $t = 0$. The probability that a non-infected contact gets traced on a given day is given by

$$P_{CT}(t) = \eta(t) \cdot \frac{N^+_{\text{PCR}}(t - \theta_{CT})}{S(t) + R(t)} = \eta_0 \cdot \frac{N^+_{\text{PCR}}(t - \theta_{CT})}{N_0}$$

where N^+_{PCR} represents the number of positive PCR tests on day $t - \theta_{CT}$, where θ_{CT} is a delay that accounts for the time it takes to trace a persons contacts (assumed to be two days).

[65]:
```
# Number of negative cases traced per positive PCR test
eta_0 <- contacts_without_covid/(64061-contacts_test_pos)

# Delay to account for time it takes to trace contacts
ct_delay <- 2
```

[76]:
```
get_pc_ct <- function(df,eta_0,delay){

  # Calculates the daily probability of being contact traced

  nday <- length(df1$day)
  N0 <- df$susceptible[1]
    +df$pre.infectious.incubation[1]
    +df$asymptomatic[1]
    +df$mild.moderate[1]
```

```
    +df$severe[1]
    +df$critical[1]
    +df$recuperation[1]
    +df$recovered[1]

  # Daily probability of a non-infected contact being traced
  p_ct <- numeric(nday)
  p_ct[1:delay] = 0
  p_ct[(delay+1):nday] = eta_0*df$test.1.pos[1:(nday-delay)]/N0

  return(p_ct)
}
```

4 Bringing it all together

In order to determine the probability of a not-infected person presenting to care, we must combine our previously derived probabilities of presenting to care due to ILI and due to contact tracing. If we assume these to be independent but non-mutually exclusive events, then we can say the following:

$$P(\text{ILI} \cap \text{CT}) = P(\text{ILI})P(\text{CT})$$
$$P(\text{ILI} \cap \text{CT}') = P(\text{ILI}) - P(\text{ILI} \cap \text{CT})$$
$$P(\text{CT} \cap \text{ILI}') = P(\text{CT}) - P(\text{ILI} \cap \text{CT})$$

Once we have calculated these probabilities, we can roll for each sucsceptible and recovered patient alive at time t to determine the number who present to care due to both ILI and CT, ILI alone, or CT alone. We do this by comparing the probabilities for each outcome to a number drawn from a uniform distribution. Those patients who present due to CT alone or ILI and CT will ultimately end up in a quarantine center after testing negative.

[8]:
```
get_extra_pc <- function(df,p_pc_ILI,eta_0,ct_delay){

  # This function combines the probabilities of being presenting to care due to␣
↪ILI or CT
  # Returns the expected number of non-infected persons presenting to care each␣
↪day

  nday <- length(df$day)

  p_ct <- get_pc_ct(df,eta_0,ct_delay)
  p_pc_ILI_and_ct <- p_ct*p_pc_ILI
  p_pc_tot <- p_pc_ILI + p_ct - p_pc_ILI_and_ct
  p_pc_only_ILI <- p_pc_ILI - p_pc_ILI_and_ct
  p_pc_only_ct <- p_ct - p_pc_ILI_and_ct

  non_infected_pop <- df$susceptible + df$recovered
```

```
  # For each non infected person on each day,
  # roll for probability of presenting to care

  df$n_pc_only_ILI <- numeric(nday)
  df$n_pc_only_ct <- numeric(nday)
  df$n_pc_ILI_and_ct <- numeric(nday)

  set.seed(123)

  for (i in seq(1,nday)){
    df$n_pc_only_ILI[i] <- sum(runif(non_infected_pop[i]) < p_pc_only_ILI[i])
    df$n_pc_only_ct[i] <- sum(runif(non_infected_pop[i]) < p_pc_only_ct[i])
    df$n_pc_ILI_and_ct[i] <- sum(runif(non_infected_pop[i]) <␣
↪p_pc_ILI_and_ct[i])
    df$n_pc_ILI_or_ct[i] <-␣
↪sum(df$n_pc_only_ILI[i]+df$n_pc_only_ct[i]+df$n_pc_ILI_and_ct[i])
  }

  return(df)
}
```

[9]:
```
# Use daily probabilities to roll for non-infected presenting to care
# This may take awhile to execute for larger simulations

df1 <- get_extra_pc(df1,p_pc_ILI_no_MSS,0,ct_delay)     # BC
df2 <- get_extra_pc(df2,p_pc_ILI_no_MSS,eta_0,ct_delay) # CT
df3 <- get_extra_pc(df3,p_pc_ILI_no_MSS,eta_0,ct_delay) # CT+IC
df4 <- get_extra_pc(df4,p_pc_ILI_MSS,eta_0,ct_delay)    # CT+IC+MSS
df5 <- get_extra_pc(df5,p_pc_ILI_no_MSS,eta_0,ct_delay) # CT+IC+QC
df6 <- get_extra_pc(df6,p_pc_ILI_MSS,eta_0,ct_delay)    # CT+IC+QC+MSS
```

5 Accounting for additional quarantine bed usage

The number of non-infected patients in quarantine centers at time t is given by

$$N^-_{QC}(t) = \sum_{i=t-14-\theta_{QC}}^{t-\theta_{QC}} \left[N^-_{\mathrm{CT}}(i) + N^-_{\mathrm{ILI}\cap\mathrm{CT}}(i)\right]$$

Where θ_{QC} represents the delay associated with the time it takes to show up to care, get a negative PCR result, and make it to a quarantine center (assumed to be 5 days). $N^-_{\mathrm{CT}}(i)$ is the number of non-infected people presenting due to contact tracing alone on day i, while $N^-_{\mathrm{ILI}\cap\mathrm{CT}}(i)$ is the number of non-infected people presenting due to contact tracing and ILI on day i. Non-infected contacts will remain in a quarantine center for 14 days before exiting.

[10]:
```
get_extra_qc <- function(df,delay){
  nday <- length(df$day)
  df$n_qc_extra <- numeric(nday)
  n_to_qc <- df$n_pc_only_ct + df$n_pc_ILI_and_ct
  df$n_qc_extra[1:delay] <- 0

  for (t in seq(1,nday)){
    startsum <- max(0,t - 14 - delay)
    endsum <- max(0,t - delay)
    df$n_qc_extra[t] <- sum(n_to_qc[startsum:endsum])
  }

  return(df)
}
```

[11]:
```
qc_delay <- 5 # Time it takes to show up to care, get negative PCR, and go to␣
↪center
```

[12]:
```
df1 <- get_extra_qc(df1,qc_delay)
df2 <- get_extra_qc(df2,qc_delay)
df3 <- get_extra_qc(df3,qc_delay)
df4 <- get_extra_qc(df4,qc_delay)
df5 <- get_extra_qc(df5,qc_delay)
df6 <- get_extra_qc(df6,qc_delay)
```

[13]:
```
# Safe dataframes as csv

outnames <- paste(fullpaths,'data_',sep="/")
outnames <- paste(outnames,myfolders,sep="")
outnames <- paste(outnames,'.csv',sep="")

write.csv(df1,outnames[1])
write.csv(df2,outnames[2])
write.csv(df3,outnames[3])
write.csv(df4,outnames[4])
write.csv(df5,outnames[5])
write.csv(df6,outnames[6])
```

[ ]:
```

```