Final Project: ABC modelling for disease outbreak

Andrea Chung

12/14/2020

Introduction

For the final project, I chose to do Approximate Bayesian Computation (ABC) computation for disease outbreak. ABC will be used to fit the parameters in a model of spread of the virus -- Influenza A and B. Bayesian approach provides probability of hypothesis of the given data. The calculation of likelihood can be removed in ABC. The ultimate goal of ABC is to approximate posterior distribution using the given prior distribution $P(\theta)$ of parameter θ.

Goal

Goal of this project is to use ABC to test the hypothesis of whether different outbreaks of same virus and outbreaks of two different virus can be presented with the same model of the spread. Through the computations, the given models by Toni and Stumpf (Figure 3a and 3c).

Approximate Bayesian Computation (ABC)

1. Sample parameter vector θ from prior distribution P.
2. Simulate the data set $D*$ by applying the sampled parameters to conditional probability distribution
3. Compare the simulated data to observed data using the the summary statistics function. If the $|\text{summary\_stat}(y*) - \text{summary\_stat}(y_0)| < \epsilon$ then keep the value of randomly chosen parameter.

Procedure

Supplementary Data

The given supplementary data tables were used to get the observed data set. There were two tables: Table 1 being same virus outbreak at different time period and Table 2 being outbreaks of two different virus at same timeframe.

Table Interpretation : The first row indicates the initial susceptibles in a household and first column indicates number of infected individuals in a household. Thus, table would be interpreted as following: 66 infected with influenza A when there was 1 initial susceptibles and 0 infected individuals.

Data 1: Influenza A infection in 1977-78

```
df_same_time1 = matrix(data = c(66, 87, 25, 22, 4,
                                13, 14, 15, 9, 4,
                                NA, 4, 4, 9, 1,
                                NA, NA, 4, 3, 1,
                                NA, NA, NA, 1, 1,
                                NA, NA, NA, NA, 0), nrow = 6,
                       ncol = 5, byrow = TRUE)

rownames(df_same_time1) <- 0:5
colnames(df_same_time1) <- 1:5
df_same_time1
```

```
##    1  2  3  4 5
## 0 66 87 25 22 4
## 1 13 14 15  9 4
## 2 NA  4  4  9 1
## 3 NA NA  4  3 1
## 4 NA NA NA  1 1
## 5 NA NA NA NA 0
```

Data 2: Influenza A infection in 1980-81

```
df_same_time2 = matrix(data = c(44, 62, 47, 38, 9,
                                10, 13, 8, 11, 5,
                                NA, 9, 2, 7, 3,
                                NA, NA, 3, 5, 1,
                                NA, NA, NA, 1, 0,
                                NA, NA, NA, NA, 1), nrow = 6,
                       ncol = 5, byrow = TRUE)

rownames(df_same_time2) <- 0:5
colnames(df_same_time2) <- 1:5
df_same_time2
```

```
##    1  2  3  4 5
## 0 44 62 47 38 9
## 1 10 13  8 11 5
## 2 NA  9  2  7 3
## 3 NA NA  3  5 1
## 4 NA NA NA  1 0
## 5 NA NA NA NA 1
```

Data 3: Influenza B infection in 1975-76

```
df_diff_A = matrix(data = c(9, 12, 18, 9, 4,
                            1, 6, 6, 4, 3,
                            NA, 2, 3, 4, 0,
                            NA, NA, 1, 3, 2,
                            NA, NA, NA, 0, 0,
                            NA,NA,NA,NA,0), nrow = 6,
                   ncol = 5, byrow= T)

rownames(df_diff_A) <- 0:5
colnames(df_diff_A) <- 1:5
df_diff_A
```

```
##    1  2  3  4 5
## 0  9 12 18  9 4
## 1  1  6  6  4 3
## 2 NA  2  3  4 0
## 3 NA NA  1  3 2
## 4 NA NA NA  0 0
## 5 NA NA NA NA 0
```

Data 4: Influenza A infection in 1978-79

```
df_diff_B = matrix(data = c(15, 12 ,4,
                            11, 17, 4,
                            NA, 21, 4,
                            NA, NA, 5), nrow = 4,
                   ncol = 3, byrow = T)

rownames(df_diff_B) <- 0:3
colnames(df_diff_B) <- 1:3
df_diff_B
```

```
##    1  2 3
## 0 15 12 4
## 1 11 17 4
## 2 NA 21 4
## 3 NA NA 5
```

Primary function - ABC sample generator

The primary function to be used will be ABC sample generator. The arguments will be _ two of observed data, summary statistics function, data generating function, and epsilon_. Through this function, the pair of parameters qh and qc will be inferred.
qc = probability of susceptible **NOT** getting infected from the community
qh = probability of suscesptible **escapes** infection from its household

- obs_data1 & obs_data2 = observed data
- sum_stat = summary statistics function
- eps = epsilon

```
abc_sample_generate = function(obs_data1, obs_data2, eps) {
  #randomly chosen parameters -- qc and qh
  q_values1 <- runif(n = 2, min = 0, max = 1)
  q_values2 <- runif(n = 2, min = 0, max = 1)
  y1 <- data_gen_fun(q_values1, obs_data1)
  y2 <- data_gen_fun(q_values2, obs_data2)
  data <- list("y1" = y1, "y2" = y2)
  sum_stat = distance(obs_data1, obs_data2, data)
  sample = while(TRUE) {
    if(sum_stat <= eps) {
    return(list(round(q_values1[1], 5), round(q_values1[2], 5),
                round(q_values2[1], 5), round(q_values2[2], 5)))
    }
  }
  return(sample)
}
```

The paramters (qc, qh) will be chosen randomly from prior distribution, which is uniform distribution range [0,1]. Then the chosen parameter will be applied to data generating function.

Data Generating Function

The data generating function will be created by using the following equation extracted from the given paper:

$$w_{js} = \binom{s}{j} w_{jj} (q_c q_h^j)^{s-j}$$

The equation computes the probability of j out of s susceptibles in household become infected.

Before simulating values into data generating function, probability matrix was created by applying the above equation. The prob_model_matrix function has the following arguments: * q_values = list of qc and qh chosen randomly from the prior distribution * data = given data / observed data

I decided to create probability matrix because the summary statistics function required computation of frobenius norm, which is computation of matrix norm.

```
prob_model_matrix = function(q_values, obs_data) {
  obs_data[is.na(obs_data) <- 0]
  s = ncol(obs_data)
  w = matrix(0, nrow = nrow(obs_data), ncol = s)

  #calculating w0s
  w[1,] <- sapply(1:s, function(n) (q_values[2])^n)

  #calculation of first column
  w[2,1] <- 1 - w[1,1]

  #calculation for the rest
  for(i in 2:s) {
    for(j in 2:nrow(w)) {
      if(j <= i) {
        w[j,i] = choose(i, j-1) * (w[j, j-1]) * ((q_values[2]) *
                                                   (q_values[1])^(j-1))^(i-j+1)
      }else{
        w[j,i] <- 1 - sum(w[,i])
        break
      }
    }
  }
  init = colSums(obs_data, na.rm = T)[1]
  prob_matrix = rmultinom(n = 1, size = init, prob = w[,1])
  for(k in 2:s) {
    next_i = colSums(obs_data, na.rm = T)[k]
    prob_calc = rmultinom(n = 1, size = next_i, prob = w[,k])
    prob_matrix <- cbind(prob_matrix, prob_calc)
  }
  return(prob_matrix)
}
```

rmultinom was used to generate multinominally distributed random number vectors and compute multiominal probabilities. Since the w_{js} has probability distribution involving two or more variables, the rmultinom was appropriate to use when generating probability matrix. The following function will be used to extract result by applying parameters (q_c and q_h).

```
data_gen_fun = function(q_values, obs_data) {
  y = prob_model_matrix(q_values, obs_data)
  return(y)
}
```

As the comparison needs to be made using two data tables (df_same_time1 and df_same_time2 && df_diff_A and df_diff_B), 2 different y's will be extracted and create a list.

```
#y1 = data_gen_fun(q_values, obs_data)
#y2 = data_gen_fun(q_values, obs_data)
#gen_data = list("y1" = y1, "y2" = y2)
```

Summary Statistics Function

Summary statistics function helps to process data by making decision whether to keep or discard the chosen

data. The summary statistics function will be created by using the equation as the following:

$$d(D_0, D^*) = \frac{1}{2}(||D_1 - D^*(q_{h1}, q_{c1})||_F + ||D_2 - D^*(q_{h2}, q_{c2})||_F)$$

The distance function is used with Frobenius norm computation. If the value from the distance is smaller than the epsilon, the theta value will be kept. Else, the values will be discarded.

```
library(matrixcalc)
distance = function(obs_data1, obs_data2, gen_data) {
  obs_data1[is.na(obs_data1)] <- 0
  obs_data2[is.na(obs_data2)] <- 0
  return(0.5 * (norm(obs_data1 - gen_data$y1, type = "F") +
                  norm(obs_data2 - gen_data$y2, type = "F")))
}
```

Posterior Distribution

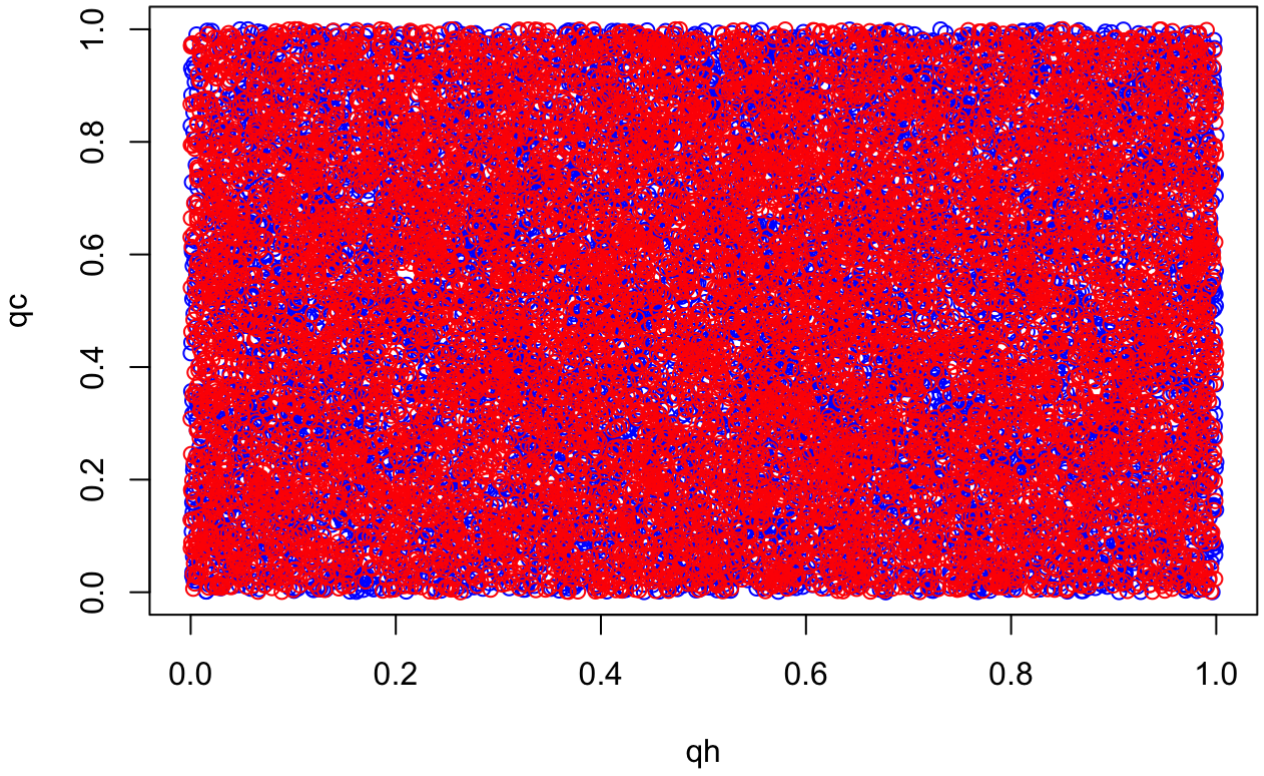
After obtaining a sample from ABC_sample generator function, the process will be repeated to create more sampled data.

```
#posterior_dist = replicate(n = 10000, abc_sample_generate(obs_data1, obs_data2, sum_sta
t, eps))
```

Plotting

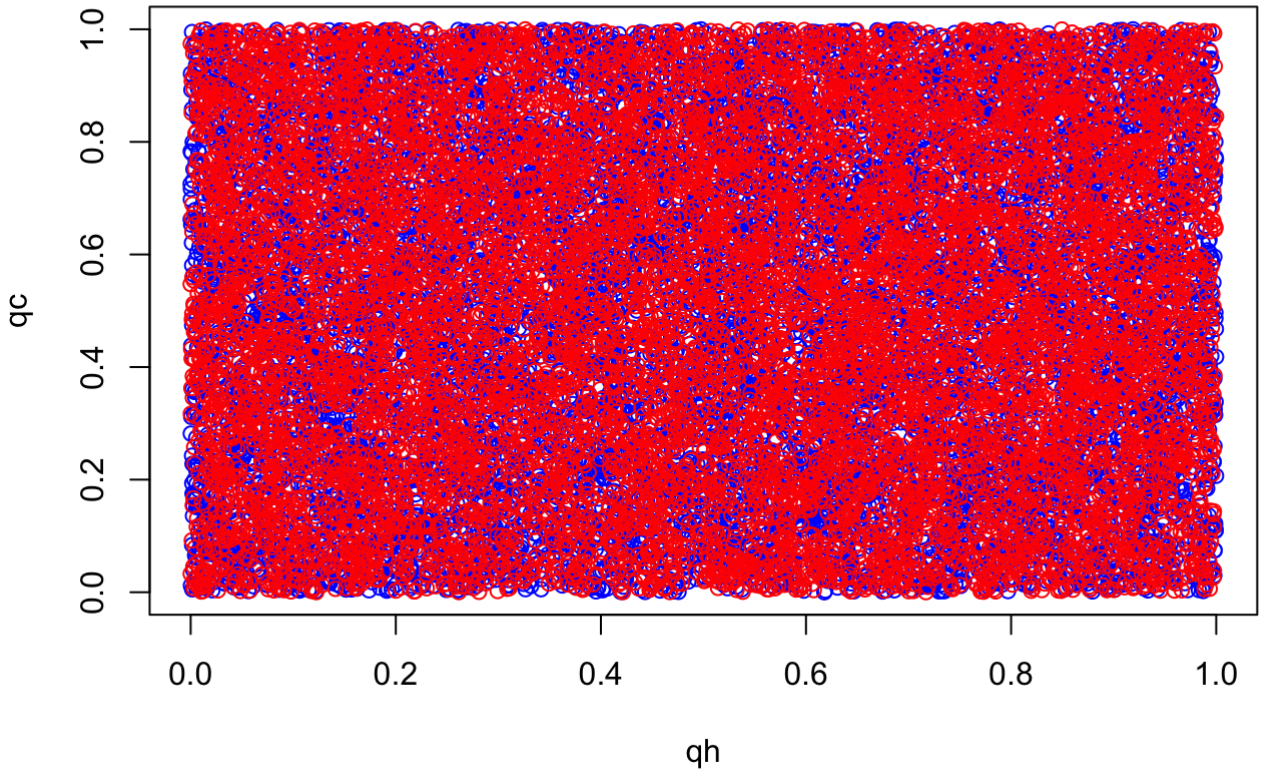
Using the posterior distribution found, I created a model inferring the relationship between qc and qh (2 parameters). Model 1 shows the q_c and q_h relationship draws different outbreaks of same virus.

```
model1 <- replicate(n = 10000,
                    abc_sample_generate(df_same_time1, df_same_time2,eps = 200))

model1 <- as.data.frame(t(model1))
rownames(model1) <- NULL
colnames(model1) <- c("qh1", "qc1", "qh2", "qc2")

plot(model1[,1], model1[,2], col = c("red"), xlim = c(0,1), ylim = c(0,1), xlab = "qh",
 ylab = "qc",
     points(model1[,3], model1[,4], col = c("blue")))
```



Model 2 shows the relationship between q_c and q_h when different virus outbreak but at the similar time period.

```
model2 = replicate(n = 10000,
                   abc_sample_generate(df_diff_A, df_diff_B,eps = 60))
model2 <- as.data.frame(t(model2))
rownames(model2) <- NULL
colnames(model2) <- c("qh1", "qc1", "qh2", "qc2")

plot(model2[,1], model2[,2], col = c("red"), xlim = c(0,1), ylim = c(0,1), xlab = "qh",
 ylab = "qc",
     points(model2[,3], model2[,4], col = c("blue")))
```



Conclusion

The plotting model is supposed to be somewhat similar to the given plot figures from Toni and Stumpf's journal. According to the result shown from the given, according to the model 1, two different outbreaks of same virus shows that both share almost the same epidemiological features. However, compared to model 1, model 2 shows that two outbreaks does not share the same features as model 1 did.

Reference

Tina Toni, Michael P. H. Stumpf, Simulation-based model selection for dynamical systems in systems and population biology, Bioinformatics, Volume 26, Issue 1, 1 January 2010, Pages 104–110, https://doi.org/10.1093/bioinformatics/btp619 (https://doi.org/10.1093/bioinformatics/btp619)