Code for Human Mobility Patterns SI

Daniel T. Citron

4/12/2019

Read in input data

"Supplement_Source_Data.csv" is a file that contains all of the data included in "Figure 6 and Figures SI" from the Source Data document.

The column names are as follows:

- `lon` & `lat` - latitude and longitude
- `travelOFF_predicted` - Travel Prevalence among all off-island travelers (TP)
- `travelEG_predicted` - Travel Prevalence among all travelers to Río Muni (TP_{rm})
- `prall_predicted` - Estimated *Pf*PR among all residents
- `prOFF_predicted` - Estimated *Pf*PR among all off-island travelers
- `prEG_predicted` - Estimated *Pf*PR among all travelers to Río Muni ($Pf\text{PR}_{\text{rm}}$)
- `prnotravel_predicted` - Estimated *Pf*PR among all non-travelers ($Pf\text{PR}_{\text{nt}}$)
- `population` - total population in that area

```
BI.data.input <- as.data.table(read.csv("Supplement_Source_Data.csv", skip = 1))
# Define total population
pop.total <- sum(BI.data.input$population)
```

Define Functions

- The function `Travel2h` co-estimates `h` and `eta` (and outputs `h` only), and does this by minimizing the difference between PR and $(h + \eta\delta)/(h + \eta\delta + r)$. Note that this does not always result in an exact solution, but we'll get as close as we can.
- The function `Travel2eta` estimates `eta`. Note that `h` defaults to 0, but that if we plug in `h` using the results from `Travel2h` we get the co-estimated value for `eta`. Note also that `eta` $\propto$ the difference between prevalence among travelers and the local residual transmission - we can only use this to calculate our co-estimate, not either the upper or lower bounds
- The function `Travel2TF` estimates the travel fraction once `eta` is known
- The function `Travel2PR` calculates $\text{PR} = (h + \eta\delta)/(h + \eta\delta + r)$

```
Travel2PR = function(TP, PRt, h=0, T=56, r=1/200, eta=NULL){
  ###
  # Solve for PR, based on delta, eta, h
  # Analogous to Equation S1
  #
  # TP = Travel Prevalence, probability of traveling in study period
  # PRt = Prevalence among travelers
  # h = Force of infection
  # T = Study period
  # r = rate of recovery
  # eta = fraction of travelers who return infected
  #
  ###
  # if eta is unknown, calculate using Travel2eta, defined below
```

```
  if(is.null(eta)) eta = Travel2eta(TP, PRt, h, T, r)
  delta = -log(1-TP)/T
  return((eta*delta + h)/(eta*delta + h + r))
}

Travel2h=function(TP, PRt, PR, T=56, r=1/200, hmx = 3/365, eta=NULL){
  ###
  # Solve for h, based on eta, PR, and PRt (among travelers)
  # Analogous to Equation S1
  #
  # TP = Travel Prevalence, probability of traveling in study period
  # PRt = Prevalence among travelers
  # PR = Prevalence among everybody
  # h = Force of infection
  # T = Study period
  # r = rate of recovery
  # hmx = maximum force of infection for search limit
  # eta = fraction of travelers who return infected
  #
  ###
  geth = function(h){
    abs(PR - Travel2PR(TP, PRt, h, T, r, eta))^2#^(1/2)
  }
  optimize(geth, c(0,hmx))$min
}

Travel2eta = function(TP, PRt, h=0, T=56, r=1/200){
  ###
  # Solve for eta, based on h (FOI) PR, and PRt (among travelers)
  # Analogous to Equation S6, for the co-estimation case
  # (other two cases can be calculated explicitly below)
  #
  # TP = Travel Prevalence, probability of traveling in study period
  # PRt = Prevalence among travelers
  # h = Force of infection
  # T = Study period
  # r = rate of recovery
  #
  ###
  LR = h/(h+r) # Local residual PR - equilibrium value when there are no importations
  # Equation 6
  prt = function(t){exp(-r*t)}
  return(T*pmax(PRt-LR,0)/integrate(prt, 0, T)$val)
}

Travel2TF = function(TP, PRt, PR, h=0, T=56, r=1/200, eta=NULL){
  ###
  # Solve for TF, based on TP, PRt (among travelers), and PR
  # Analogous to Equation S3
  #
  # TP = Travel Prevalence, probability of traveling in study period
  # PRt = Prevalence among travelers
  # h = Force of infection
```

```
  # T = Study period
  # r = rate of recovery
  # eta
  #
  ###
  tpr = Travel2PR(TP, PRt, 0, T, r, eta)
  pmin(tpr/PR, 1)
}
```

Rate of travel

Solve for `delta` using Equation 5 from the Supplemental Information. Do this using the TP for both all off-island travel and those who traveled to Río Muni.

```
BI.data.input$delta.all <- -log(1 - BI.data.input$travelOFF_predicted)/56
BI.data.input$delta.rm <- -log(1 - BI.data.input$travelEG_predicted)/56
```

Define rate of recovery

```
r = 1/200.
```

Calculate Probability of Returning from Travel with Infection

We calculate the upper, lower, and coestimated values for `eta` using Equation 6 from the Supplemental Information. We also calculate the force of infection (`h`) at the same time.

First, for all off-island travelers (which we use to generate Figure S3):

```
# upper
BI.data.input$eta.upper <- pmax((BI.data.input$prOFF_predicted),0)*56*r/(1 - exp(-56*r))
BI.data.input$h.upper <- 0
for (i in 1:194){
  BI.data.input$h.upper[[i]] <- Travel2h(TP = BI.data.input$travelOFF_predicted[[i]],
                                         PRt = BI.data.input$prOFF_predicted[[i]],
                                         PR = BI.data.input$prall_predicted[[i]],
                                         eta = BI.data.input$eta.upper[[i]])
}

# lower
BI.data.input$eta.lower <- pmax((BI.data.input$prOFF_predicted -
                                   BI.data.input$prall_predicted),0)*56*r/(1 - exp(-56*r))
BI.data.input$h.lower <- 0
for (i in 1:194){
  BI.data.input$h.lower[[i]] <- Travel2h(TP = BI.data.input$travelOFF_predicted[[i]],
                                         PRt = BI.data.input$prOFF_predicted[[i]],
                                         PR = BI.data.input$prall_predicted[[i]],
                                         eta = BI.data.input$eta.lower[[i]])
}
```

```r
# Coestimation - goes into Supplementary Figure 3
BI.data.input$h.co <- 0
for (i in 1:194){
  BI.data.input$h.co[[i]] <- Travel2h(BI.data.input$travelOFF_predicted[[i]],
                                      BI.data.input$prOFF_predicted[[i]],
                                      BI.data.input$prall_predicted[[i]])
}
BI.data.input$eta.co <- 0
for (i in 1:194){
  BI.data.input$eta.co[[i]] <- Travel2eta(BI.data.input$travelOFF_predicted[[i]],
                                          BI.data.input$prOFF_predicted[[i]],
                                          h = BI.data.input$h.co[[i]])
}
```

Calculate the same quantities for all travelers to Río Muni (which we use to generate Figures S1 and S2):

```r
# upper
BI.data.input$eta.upper.rm <- pmax((BI.data.input$prEG_predicted),0)*56*r/(1 - exp(-56*r))
BI.data.input$h.upper.rm <- 0
for (i in 1:194){
  BI.data.input$h.upper.rm[[i]] <- Travel2h(TP = BI.data.input$travelEG_predicted[[i]],
                                            PRt = BI.data.input$prEG_predicted[[i]],
                                            PR = BI.data.input$prall_predicted[[i]],
                                            eta = BI.data.input$eta.upper.rm[[i]])
}
# lower
BI.data.input$eta.lower.rm <- pmax((BI.data.input$prEG_predicted -
                                  BI.data.input$prall_predicted),0)*56*r/(1 - exp(-56*r))
BI.data.input$h.lower.rm <- 0
for (i in 1:194){
  BI.data.input$h.lower.rm[[i]] <- Travel2h(TP = BI.data.input$travelEG_predicted[[i]],
                                            PRt = BI.data.input$prEG_predicted[[i]],
                                            PR = BI.data.input$prall_predicted[[i]],
                                            eta = BI.data.input$eta.lower.rm[[i]])
}
# Coestimation - goes into Figure 6
BI.data.input$h.co.rm <- 0
for (i in 1:194){
  BI.data.input$h.co.rm[[i]] <- Travel2h(TP = BI.data.input$travelEG_predicted[[i]],
                                         PRt = BI.data.input$prEG_predicted[[i]],
                                         PR = BI.data.input$prall_predicted[[i]])
}
BI.data.input$eta.co.rm <- 0
for (i in 1:194){
  BI.data.input$eta.co.rm[[i]] <- Travel2eta(TP = BI.data.input$travelEG_predicted[[i]],
                                             PRt = BI.data.input$prEG_predicted[[i]],
                                             h = BI.data.input$h.co.rm[[i]])
}
```

Calculate Travel Fraction

We calculate the Travel Fraction (TF) based on all off-island travelers

```r
# Coestimate TF - Goes into Supplementary Figure 3
BI.data.input$tf.co = rep(0, 194)
for (i in 1:194){
  BI.data.input$tf.co[[i]] = Travel2TF(TP = BI.data.input$travelOFF_predicted[[i]],
                                       PRt = BI.data.input$prOFF_predicted[[i]],
                                       PR = BI.data.input$prall_predicted[[i]],
                                       eta=BI.data.input$eta.co[[i]])
}
# Upper TF
BI.data.input$tf.upper <- with(BI.data.input,
                               pmin((eta.upper*delta.all)/(eta.upper*delta.all +
                                                           1/200)/prall_predicted,1))
# Lower TF
BI.data.input$tf.lower <- with(BI.data.input,
                               pmin((eta.lower*delta.all)/(eta.lower*delta.all +
                                                           1/200)/prall_predicted,1))
```

We calculate the Travel Fraction (TF) based on travelers to Río Muni

```r
# Coestimate TF - Goes into Figure 6
BI.data.input$tf.co.rm = rep(0, 194)
for (i in 1:194){
  BI.data.input$tf.co.rm[[i]] = Travel2TF(TP = BI.data.input$travelEG_predicted[[i]],
                                         PRt = BI.data.input$prEG_predicted[[i]],
                                         PR = BI.data.input$prall_predicted[[i]],
                                         eta = BI.data.input$eta.co.rm[[i]])
}
# Upper TF
BI.data.input$tf.upper.rm <- with(BI.data.input,
                                 pmin((eta.upper.rm*delta.rm)/(eta.upper.rm*delta.rm +
                                                               1/200)/prall_predicted,1))
# Lower TF
BI.data.input$tf.lower.rm <- with(BI.data.input,
                                 pmin((eta.lower.rm*delta.rm)/(eta.lower.rm*delta.rm +
                                                               1/200)/prall_predicted,1))
```

Generate Tables

How many people across all of Bioko live in areas where the travel fraction is high?

For travelers to Río Muni (which we show in Table S1)

```r
# Coestimate
print("Coestimate")
```

```
## [1] "Coestimate"
```

```r
c(sum(BI.data.input[tf.co.rm >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.co.rm >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.co.rm >= 0.5]$pop)/pop.total)
```

```
## [1] 0.4835583 0.6721487 0.7476300
```

```r
# Upper
print("Upper")
```

```
## [1] "Upper"
```

```r
c(sum(BI.data.input[tf.upper.rm >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.upper.rm >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.upper.rm >= 0.5]$pop)/pop.total)
```

```
## [1] 0.5825050 0.7092454 0.8346979
```

```r
# Lower
print("Lower")
```

```
## [1] "Lower"
```

```r
c(sum(BI.data.input[tf.lower.rm >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.lower.rm >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.lower.rm >= 0.5]$pop)/pop.total)
```

```
## [1] 0.09762337 0.27202036 0.62692631
```

For all travelers (which we show in Table S2):

```r
# Coestimate
print("Coestimate")
```

```
## [1] "Coestimate"
```

```r
c(sum(BI.data.input[tf.co >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.co >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.co >= 0.5]$pop)/pop.total)
```

```
## [1] 0.3797212 0.5193494 0.7222616
```

```r
# Upper
print("Upper")
```

```
## [1] "Upper"
```

```r
c(sum(BI.data.input[tf.upper >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.upper >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.upper >= 0.5]$pop)/pop.total)
```

```
## [1] 0.5357734 0.6972250 0.7797212
```

```r
# Lower
print("Lower")
```

```
## [1] "Lower"
```

```r
c(sum(BI.data.input[tf.lower >= 1]$pop)/pop.total,
  sum(BI.data.input[tf.lower >= 0.8]$pop)/pop.total,
  sum(BI.data.input[tf.lower >= 0.5]$pop)/pop.total)
```

```
## [1] 0.09985395 0.10741314 0.40901527
```

Sensitivity to Treatment

How does the model change when we add treatment? This enters into our mechanistic model by reducing the mean duration of infections. We recalculate `eta` and `h` for the Río Muni case, this time with `r`= 1/100.

```r
# upper
BI.data.input$eta.upper.r <-pmax((BI.data.input$prEG_predicted),0)*56/100/(1-exp(-56/100))
BI.data.input$h.upper.r <- 0
for (i in 1:194){
  BI.data.input$h.upper.r[[i]] <- Travel2h(BI.data.input$travelEG_predicted[[i]],
                                          BI.data.input$prEG_predicted[[i]],
                                          BI.data.input$prall_predicted[[i]],
                                          r = 1/100,
                                          eta = BI.data.input$eta.upper.r[[i]])
}
# lower
BI.data.input$eta.lower.r <- pmax((BI.data.input$prEG_predicted -
                                  BI.data.input$prall_predicted),0)*56/100/(1-exp(-56/100))
BI.data.input$h.lower.r <- 0
for (i in 1:194){
  BI.data.input$h.lower.r[[i]] <- Travel2h(BI.data.input$travelEG_predicted[[i]],
                                          BI.data.input$prEG_predicted[[i]],
                                          BI.data.input$prall_predicted[[i]],
                                          r = 1/100,
                                          eta = BI.data.input$eta.lower.r[[i]])
}
# coestimate
BI.data.input$h.co.r <- 0
for (i in 1:194){
  BI.data.input$h.co.r[[i]] <- Travel2h(BI.data.input$travelEG_predicted[[i]],
                                       BI.data.input$prEG_predicted[[i]],
                                       BI.data.input$prall_predicted[[i]],
                                       r = 1/100,
                                       hmx = 1)
}
BI.data.input$eta.co.r <- 0
for (i in 1:194){
  BI.data.input$eta.co.r[[i]] <- Travel2eta(BI.data.input$travelEG_predicted[[i]],
                                           BI.data.input$prEG_predicted[[i]],
                                           h = BI.data.input$h.co.r[[i]],
                                           r = 1/100)
}
BI.data.input$tf.co.r <- 0
for (i in 1:194){
  BI.data.input$tf.co.r[[i]] <- Travel2TF(BI.data.input$travelEG_predicted[[i]],
                                         BI.data.input$prEG_predicted[[i]],
                                         BI.data.input$prall_predicted[[i]],
                                         h = BI$h.co.r[[i]],
                                         r = 1/100,
                                         eta = BI.data.input$eta.co.r[[i]])
}

## Plotting Figure S4:
# plot(with(BI.data.input, h.co.rm/(h.co.rm + 1/200)),
#      with(BI.data.input, h.co.r/(h.co.r + 1/100))
# )
# segments(0,0,1,1)
#
```

```
# plot(with(BI.data.input, tf.co.rm),
#      with(BI.data.input, tf.co.r)
#      )
# segments(0,0,1,1)
```

Sensitivity to Travel Heterogeneity

What happens when travel behavior is no longer assumed to be distributed homogeneously among all people? We ignore local transmission and consider cases where we redistribute travel behavior such that half of people in each area never travel while the other half travel twice as much.

This first requires re-calculating `delta` in the heterogeneous case:

```
fac = .5
BI.data.input$delta.h = -log(1 - BI.data.input$travelEG_predicted/fac)/56
```

We then also calculate the new *Pf*PR for all people, using SI Equation 1 with the new values for `delta` for half of the travelers, as described in SI Section 3.3:

```
BI.data.input$pr_h <- with(BI.data.input, delta.h*eta.co.rm/(delta.h*eta.co.rm + r)*fac)

## Plotting Figure S5:
# plot(with(BI.data.input, delta.rm*eta.co.rm/(delta.rm*eta.co.rm + r)),
#      with(BI.data.input, delta.h*eta.co.rm/(delta.h*eta.co.rm + r)*fac))
# segments(0,0,1,1)
```