Environ Health Perspect

DOI: 10.1289/EHP12314

Supplemental Material

Opportunities to Interrupt Transmission of Enteropathogens of Poultry Origin in Maputo, Mozambique: A Transmission Model Analysis

Kayoko Shioda, Andrew F. Brouwer, Frederica Lamar, Hermógenes N. Mucache, Karen Levy, and Matthew C. Freeman

Table of Contents

Table S1. The estimated proportion of illnesses caused by each transmission pathway for the AFR E region, reported in the WHO FERG report 2007-2015.

Table S2. Changes in the simulated steady-state prevalence of *Campylobacter* infection after intervention (50% reduction in each transmission rate).

Table S3. Changes in the simulated steady-state prevalence of *Salmonella* infection after intervention (50% reduction in each transmission rate).

Figure S1. Prior and posterior distributions of the prevalence of total infection and contribution of each pathway to the force of infection for *Campylobacter* (A) and *Salmonella* (B).

Figure S2. Changes in the simulated steady-state prevalence of total infection of *Shigella* (*I*) after reducing the transmission rate for each pathway.

R script 1: campy.R

R script 2: salm.R

Table S1. The estimated proportion of illnesses caused by each transmission pathway for the AFR E region, reported in the WHO FERG report 2007-2015.

Transmission pathway	Proportion of illnesses caused by each pathway (median and 95% uncertainty interval)	
	Campylobacter	**Non-typhoidal *Salmonella***
Person-to-person	4% (0-23%)	18% (0-48%)
Water	9% (1-29%)	10% (0-40%)
Food	57% (29-77%)	46% (10-73%)
Animal	17% (0-42%)	15% (0-42%)
Soil	0% (0-12%)	1% (0-19%)
All other*	6% (0-16%)	2% (0-8%)

Data in this table can be found in the following report:
WHO | WHO estimates of the global burden of foodborne diseases. WHO. Accessed September 16, 2020. http://www.who.int/foodsafety/publications/foodborne_disease/fergreport/en/

Transmission models were fit to the median values.

*For the all-other pathway, we did not directly use the proportion reported in the WHO FERG report, because if we use the reported values directly, proportions did not add up to 1. Therefore, we subtracted the sum of the proportions for person-to-person, water, food, animal, and soil pathways from 100% and used that as a proportion for the all-other pathway. E.g., for *Campylobacter*, the proportion of illnesses caused by all-other pathway was 100 – (4+9+57+17+0) = 13%.

Table S2. Changes in the simulated steady-state prevalence of *Campylobacter* infection after intervention (50% reduction in each transmission rate)

Intervention scenario	Changes in the simulated steady-state prevalence of *Campylobacter* infection after intervention					
	Total infection	Infection via person-to-person	Infection via water	Infection via food	Infection via live animals	Infection via all-other sources
Reducing the rate of person-to-person transmission by 50%	-2.6% (-6.8%, -0.7%)	-51.2% (-53.1%, -50.3%)	-2.4% (-6.3%, -0.6%)	0.2% (0.1%, 0.6%)	0.2% (0.1%, 0.6%)	0.2% (0.1%, 0.6%)
Reducing the rate of waterborne transmission by 50%	-5.2% (-10.5%, -2%)	-4.8% (-9.6%, -1.8%)	-52.4% (-54.8%, -50.9%)	0.5% (0.2%, 1%)	0.5% (0.2%, 1%)	0.5% (0.2%, 1%)
Reducing the rate of foodborne transmission by 50%	-28.4% (-34.5%, -21.6%)	-26.5% (-32.6%, -20.1%)	-26.5% (-32.6%, -20.1%)	-48.8% (-49.1%, -48.3%)	2.5% (1.7%, 3.4%)	2.5% (1.7%, 3.4%)
Reducing the rate of live animal transmission by 50%	-10% (-15.7%, -5.6%)	-9.2% (-14.6%, -5.1%)	-9.2% (-14.6%, -5.1%)	0.9% (0.5%, 1.5%)	-49.6% (-49.8%, -49.3%)	0.9% (0.5%, 1.5%)
Reducing the rate of all-other transmission by 50%	-7.8% (-13.2%, -3.7%)	-7.1% (-12.2%, -3.4%)	-7.1% (-12.2%, -3.4%)	0.7% (0.3%, 1.2%)	0.7% (0.3%, 1.2%)	-49.7% (-49.8%, -49.4%)

Table S3. Changes in the simulated steady-state prevalence of *Salmonella* infection after intervention (50% reduction in each transmission rate)

Intervention scenario	Changes in the simulated steady-state prevalence of *Salmonella* infection after intervention						
	Total infection	**Infection via person-to-person**	**Infection via water**	**Infection via food**	**Infection via live animals**	**Infection via soil**	**Infection via all-other sources**
Reducing the rate of person-to-person transmission by 50%	-8.8% (-12.4%, -6.3%)	-53.3% (-54.8%, -52.4%)	-6.7% (-9.6%, -4.7%)	2.4% (1.6%, 3.4%)	2.4% (1.6%, 3.4%)	-6.7% (-9.6%, -4.7%)	2.4% (1.6%, 3.4%)
Reducing the rate of waterborne transmission by 50%	-5.1% (-8.3%, -3.1%)	-3.9% (-6.3%, -2.3%)	-51.9% (-53.2%, -51.1%)	1.4% (0.8%, 2.2%)	1.4% (0.8%, 2.2%)	-3.9% (-6.3%, -2.3%)	1.4% (0.8%, 2.2%)
Reducing the rate of foodborne transmission by 50%	-24.2% (-28.2%, -20.2%)	-19.4% (-23%, -15.6%)	-19.4% (-23%, -15.6%)	-46.8% (-47.4%, -46.1%)	6.4% (5.2%, 7.8%)	-19.4% (-23%, -15.6%)	6.4% (5.2%, 7.8%)
Reducing the rate of live animal transmission by 50%	-8.5% (-11.7%, -5.7%)	-6.5% (-9%, -4.3%)	-6.5% (-9%, -4.3%)	2.2% (1.5%, 3.2%)	-48.9% (-49.3%, -48.4%)	-6.5% (-9%, -4.3%)	2.2% (1.5%, 3.2%)
Reducing the rate of soil transmission by 50%	-0.8% (-2.1%, -0.1%)	-0.6% (-1.6%, -0.1%)	-0.6% (-1.6%, -0.1%)	0.2% (0%, 0.6%)	0.2% (0%, 0.6%)	-50.3% (-50.8%, -50.1%)	0.2% (0%, 0.6%)
Reducing the rate of all-other transmission by 50%	-5.6% (-8.6%, -3.2%)	-4.1% (-6.5%, -2.4%)	-4.1% (-6.5%, -2.4%)	1.5% (0.8%, 2.3%)	1.5% (0.8%, 2.3%)	-4.1% (-6.5%, -2.4%)	-49.3% (-49.6%, -48.8%)

Figure S1. Prior and posterior distributions of the prevalence of total infection and contribution of each pathway to the force of infection for *Campylobacter* (A) and *Salmonella* (B).

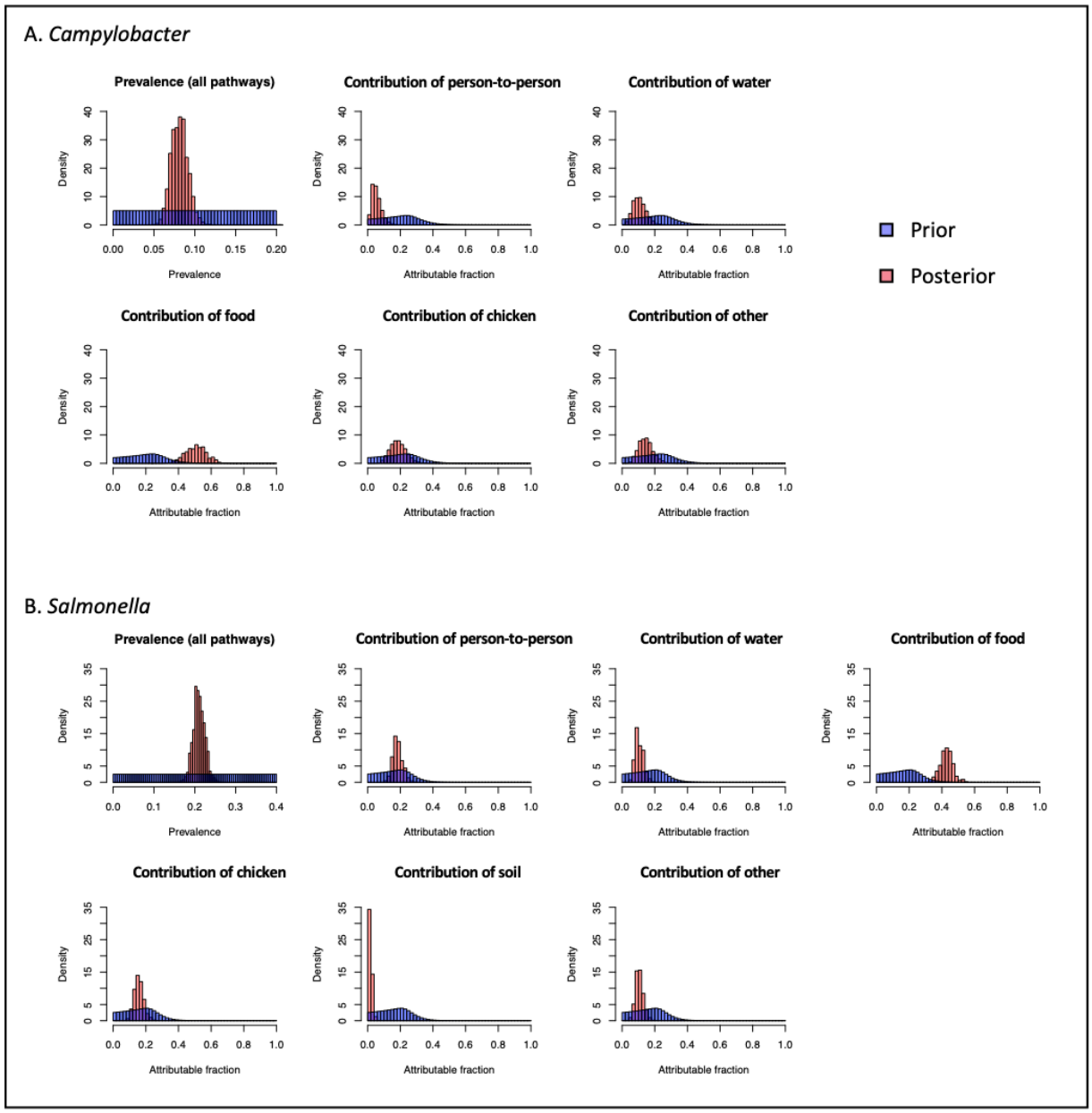
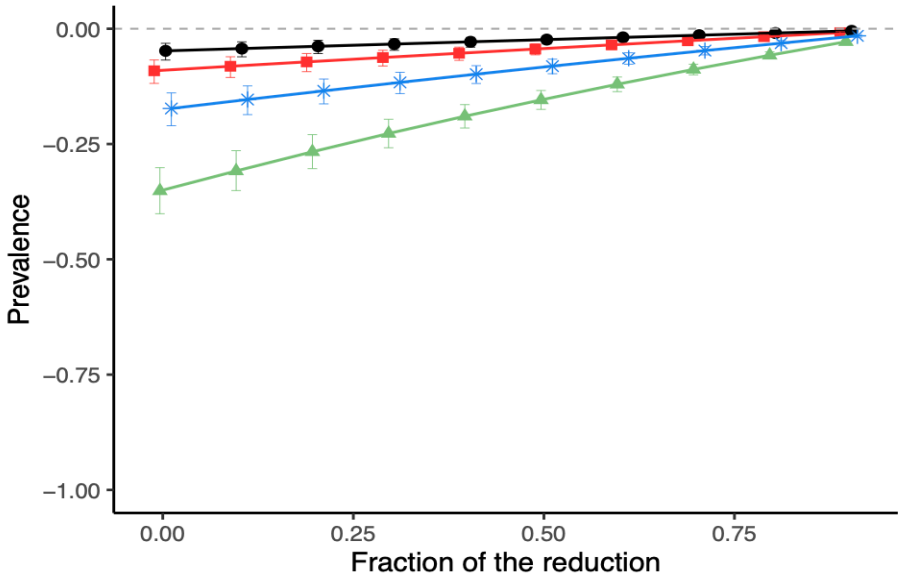


Figure S2: Changes in the simulated steady-state prevalence of total infection of *Shigella* (*I*) after reducing the transmission rate for each pathway.



Black: all-other pathway; red: food pathway; blue: water pathway; green: person-to-person pathway.

We built a susceptible-infectious-susceptible type model for *Shigella* that captures a comprehensive set of transmission pathways, namely person-to-person (direct physical contact with infected people), water-to-person (drinking water and stored water), food-to-person (chicken meat, eggs, and contaminated raw product), and all-other-sources-to-person (anything that was not captured by the aforementioned pathways).

$$\frac{dI_{hh}}{dt} = \beta_I\, S(I_{hh} + I_{hw} + I_{hf} + I_{ho}) - \gamma I_{hh} - \mu I_{hh}$$

$$\frac{dI_{hw}}{dt} = \beta_W\, SW - \gamma I_{hw} - \mu I_{hw}$$

$$\frac{dI_{hf}}{dt} = \beta_F\, SF - \gamma I_{hf} - \mu I_{hf}$$

$$\frac{dI_{ho}}{dt} = \beta_O\, SO - \gamma I_{ho} - \mu I_{ho}$$

$$\frac{dW}{dt} = \alpha(I_{hh} + I_{hw} + I_{hf} + I_{ho}) - \epsilon W$$

The model tracks the fractions of children who are susceptible to infection (S) and children who are infected via direct contact with other infected people (I_{hh}), consumption of contaminated food (I_{hf}) and water (I_{hw}), and all other exposure sources (I_{ho}) on day t. The model also includes water compartment (W) that tracks pathogen concentration in the water, which is affected by the fraction of infected individuals on day t through the rate of pathogen shedding from infected people to water (α_W) and the rate of natural pathogen clearance from water (ϵ_W) (Table 1). Pathogen contamination in food (F) and all-other sources (O) and the prevalence of infection among live animals (A) were constant and not time-varying. The mortality rate (μ) was fixed to at the reported rate among children under five years of age in Mozambique (Table 1). The birth rate (b) was also fixed and set to be the same as the mortality rate to generate a stable population over the simulation period.

R script 1: campy.R

```
################################################################################
#                                                                              #
#   PROJECT: Opportunities to interrupt transmission of enteropathogens of     #
#            poultry origin in Maputo, Mozambique: a transmission model analysis #
#                                                                              #
#      DATE: May 2023                                                          #
#  CODED BY: Kayoko Shioda, PhD, DVM, MPH                                      #
#                                                                              #
################################################################################


#------------------------------------------------------------------------------#
# Description of this R script
#------------------------------------------------------------------------------#

# Objective:
# To conduct sampling-importance resampling for the transmission dynamic model
# for Campylobacter that has 5 pathways (person-to-person, water, food,
# live-animals, all other)

#------------------------------------------------------------------------------#
# Load packages and functions and other settings
#------------------------------------------------------------------------------#

# Packages
library(lhs)
library(deSolve) # For ODE (deterministic model)
library(ggplot2)
library(SobolSequence)

# Differential equations
SIWFCOS_model <- function(t,       # Time
                          x,       # Vector containing the initial condition
                          parms) { # Vector storing transmission rates, etc.

  # Initial condition
  S   <- x[1]
  Ihh <- x[2] # Number of individuals infected by other infected individuals
  Ihw <- x[3] # Number of individuals infected by water
  Ihf <- x[4] # Number of individuals infected by food
  Ihc <- x[5] # Number of individuals infected by live chickens
  Iho <- x[6] # Number of individuals infected by other sources
  W   <- x[7] # Pathogens in water
  F   <- x[8] # Pathogens in food
  C   <- x[9] # Prevalence among live chickens
  O   <- x[10] # Pathogens in "other:
  N   <- S + (Ihh + Ihw + Ihf + Ihc + Iho)

  # Define each rate moving from one compartment to the other
  gamma  <- parms[6] # Recovery rate
  alpha  <- parms[7] # Shedding rate into water
  xi     <- parms[8] # Clearance rate from water
  mu     <- parms[9] # Birth and death rate

  # Scale transmission rates by gamma+mu
  beta_I <- parms[1]*(gamma+mu)/(Ihh+Ihw+Ihf+Ihc+Iho) # Transmission rate from person to person
  beta_W <- parms[2]*(gamma+mu)/W # Transmission rate from water to person
  beta_F <- parms[3]*(gamma+mu)/F # Transmission rate from food to person
  beta_C <- parms[4]*(gamma+mu)/C # Transmission rate from live chicken to person
  beta_O <- parms[5]*(gamma+mu)/O # Transmission rate from live chicken to person

  dxdt <- numeric(length(x))
  dxdt[1] <- mu*N -beta_I*S*(Ihh+Ihw+Ihf+Ihc+Iho) - beta_W*S*W - beta_F*S*F - beta_C*S*C -
beta_O*S*O - mu*S + gamma*(Ihh+Ihw+Ihf+Ihc+Iho) # dS/dt
  dxdt[2] <-       beta_I*S*(Ihh+Ihw+Ihf+Ihc+Iho) - gamma*Ihh - mu*Ihh # dIhh/dt (person-to-
person transmission)
  dxdt[3] <-       beta_W*S*W                     - gamma*Ihw - mu*Ihw # dIhw/dt (water-to-person
transmission)
```

```
  dxdt[4] <-       beta_F*S*F                     - gamma*Ihf - mu*Ihf # dIhf/dt (food-to-person
transmission)
  dxdt[5] <-       beta_C*S*C                     - gamma*Ihc - mu*Ihc # dIhc/dt (live chicken-
to-person transmission)
  dxdt[6] <-       beta_O*S*O                     - gamma*Iho - mu*Iho # dIho/dt (all other-to-
person transmission) <-- NEW!!
  dxdt[7] <-       alpha*(Ihh+Ihw+Ihf+Ihc+Iho) - xi*W # dW/dt (Pathogen shedding into water and
clearance from water)
  dxdt[8] <- 0 # dF/dt
  dxdt[9] <- 0 # dC/dt
  dxdt[10]<- 0 # dO/dt

  list(dxdt)
}



#-------------------------------------------------------------------------------#
# Sobol sampling for the prevalence and contribution of each pathway
#-------------------------------------------------------------------------------#

# Number of parameter sets you want to run
numsamples <-  2.5e+06

# Number of parameters you want to estiamte
numparam <- 6 # 6 for the Campylobacter model and 7 for the Salmonella model

# Sobol sampling
set.seed(721)
p <- sobolSequence.points(numparam, count=numsamples)

# Lower and upper bounds for the Latin Hypercubic sampling
lhs_low  <-  rep(0, numparam)
lhs_high <-  c(0.2, rep(1, numparam-1))

# Scale Sobol samples, using the lower and upper bounds defined above
rho <- t(lhs_low + (lhs_high - lhs_low)*t(p)) # Scaled parameter sets

# Need to have these scaled sampled attributable fractions sum to 1.
for (i in 1:numsamples){
  rho[i, 2:numparam] <- rho[i, 2:numparam]/sum(rho[i,2:numparam])
}


#-------------------------------------------------------------------------------#
# Set up parameters
#-------------------------------------------------------------------------------#

# Population size in the simulated population
N <- 1 # Should be 1, as we are modeling the prevalence

# Fixed parameters
gamma <- 1  # Recovery rate
alpha <- 1  # Shedding rate into water
xi    <- 1  # Clearance rate from water (Stays in the environment for 1/xi days)
b     <- 70/1000/365.25 * N
mu    <- 70/1000/365.25      # Background death rate for children <5 yo (70 per 1,000 population
during a year at midyear of 2019)

# Set the initial condition
# (i.e., fraction of individuals and pathogen concentration in each status at Time 1)
S   <- N - 0.1
Ihh <- 0
Ihw <- 0
Ihf <- 0
Ihc <- 0
Iho <- 0.1
W   <- 0.1 # Pathogen concentration in water, W_t
F   <- 1 # Pathogen concentration in food, F_t is constant over time (set to 1 so that it is
basically included in beta_F)
```

```
C   <- 1 # Prevalence of infected live chicken which is constant over time (set to 1 so that it
is basically included in beta_C)
O   <- 1 # Pathogen concentration in all other sources is constant over time (set to 1 so that it
is basically included in beta_O)
set.ODEtime <- seq(from=0, to=365.25*2, by=1) # Run the model for two years

# Create a vector containing the initial condition
set.x0        <- c(S[1], Ihh[1], Ihw[1], Ihf[1], Ihc[1],  Iho[1], W[1], F[1], C[1], O[1])
names(set.x0) <- c('S', 'Ihh',  'Ihw',  'Ihf',  'Ihc',   'Iho',  'W',  'F',  'C',  'O')


#-------------------------------------------------------------------------------#
# Run ODE with each parameter set and calculate nLL
#-------------------------------------------------------------------------------#

# These are containers that we'll fill in
res.prev  <- matrix(NA, nrow=numsamples, ncol=(numparam)) # estimated prevalence (prev_I,
prev_Ihh, prev_Ihw, prev_Ihf, prev_Ihc, prev_Iho) will be stored here
parsample <- matrix(NA, nrow=numsamples, ncol=(numparam)) # scaled Sobol sampled parameter sets
(prevalence and AFs) will be stored here
nllsample <- numeric(numsamples) # netative log-likelihood will be stored here
env.conc <- numeric(numsamples) # pathogen concentration in the environmental source (water in
this case) in the endemic status

# Create a function to run ODE and calculate negative log-likelihood
ode_func <- function (lambda_gm, # prevalence (all pathways) scaled by gamma*mu
                      partition) { # AFs
  out <- ode(y = set.x0,              # initial state values
             t = set.ODEtime,         # time step (t=0,1,2,3,...)
             func   = SIWFCOS_model,  # ODE function
             parms  = c(lambda_gm*partition, gamma, alpha, xi, mu), # Sobol samples and fixed
parameters
             method = 'vode')         # Method that performs integration (lsode, ode45, vode)
  out <- as.data.frame(out)

  # Calculate the proportion of infected children (combining all transmission routes)
  out$I <- out$Ihh + out$Ihw + out$Ihf + out$Ihc + out$Iho

  # Simulated prevalence at the stable stage (equilibrium)
  prev_Ihh <- tail(out[,"Ihh"],1)
  prev_Ihw <- tail(out[,"Ihw"],1)
  prev_Ihf <- tail(out[,"Ihf"],1)
  prev_Ihc <- tail(out[,"Ihc"],1)
  prev_Iho <- tail(out[,"Iho"],1)
  prev_I   <- tail(out[,"I"],  1)
  W_end    <- tail(out[,"W"],  1)

  # Calculate the negative log-likelihood using the MapSan data and the WHO FERG
  # attributable fractions (Multinomial distribution)
  NLL = -sum(759*c(0.08*c(0.04,0.09,0.57,0.17,0.13), 1-0.08) * log(c(prev_Ihh, prev_Ihw,
prev_Ihf, prev_Ihc, prev_Iho, 1-prev_I)))

  # Output
  return(c(NLL, prev_I, prev_Ihh, prev_Ihw, prev_Ihf, prev_Ihc, prev_Iho, W_end))
}

# Run ODE and calculate negative log-likelihood for each sampled parameter set
for(i in 1:numsamples){
  if (i%%1000==0) {
    print(i)
  }

  # Scaled Sobol sampled prevalence of infection
  target_prev <- rho[i,1]

  # Scale the prevalence at steady state by (gamma + mu)
  lambda_gm <- target_prev/(1-target_prev)

  # Run ODE
  temp <- ode_func(lambda_gm, # Prevalence scaled by gamma+mu
                   rho[i,-1]) # Scaled Sobol sampled AFs
```

```
  # Save outputs
  nllsample[i]  <- temp[1]  # negative log-likelihood
  parsample[i,] <- rho[i,]  # scaled Sobol sampled parameter sets (prevalence and AFs)
  res.prev[i,]  <- temp[2:(numparam+1)] # estimated prevalence (prev_I, prev_Ihh, prev_Ihw,
prev_Ihf, prev_Ihc, prev_Iho)
  env.conc[i]  <- temp[numparam+2]
}

parsample <- as.data.frame(parsample)
names(parsample) <- c("baseline.prev", "AF_h", "AF_w", "AF_f", "AF_c", "AF_o")
res.prev <- as.data.frame(res.prev)
names(res.prev) <- c("prev_I", "prev_Ihh", "prev_Ihw", "prev_Ihf", "prev_Ihc", "prev_Iho")


#-------------------------------------------------------------------------------#
# Resampling
#-------------------------------------------------------------------------------#

# Calculate the relative negative log-likelihood
rel.nllsample <- nllsample - min(nllsample)

# Calculate the probability (weights) based on the relative nLL
prob.nllsample <- exp(-rel.nllsample)

# Set the number of samples you want to resample
num.resample <- 10000

# Resample the rows based on the sample importance, prob.nllsample (with replacement)
resample.ind <- sample(1:length(nllsample), num.resample, replace=T, prob=prob.nllsample)
resampled.parsample <- parsample[resample.ind,]
resampled.res.prev  <- res.prev[resample.ind,]
resampled.env.conc  <- env.conc[resample.ind]
```

R script 2: salm.R

```
##############################################################################
#                                                                            #
#   PROJECT: Opportunities to interrupt transmission of enteropathogens of   #
#            poultry origin in Maputo, Mozambique: a transmission model analysis #
#                                                                            #
#                                                                            #
#      DATE: May 2023                                                        #
#  CODED BY: Kayoko Shioda, PhD, DVM, MPH                                    #
#                                                                            #
##############################################################################


#----------------------------------------------------------------------------#
# Description of this R script
#----------------------------------------------------------------------------#

# Objective:
# To conduct sampling-importance resampling for the transmission dynamic model
# for Campylobacter that has 5 pathways (person-to-person, water, food,
# live-animals, all other)

#----------------------------------------------------------------------------#
# Load packages and functions and other settings
#----------------------------------------------------------------------------#

# Packages
library(lhs)
library(deSolve)
library(ggplot2)
library(SobolSequence)

# Differential equations
SIWFCEOS_model <- function(t,       # Time
                           x,       # Vector containing the initial condition
                           parms) { # Vector storing transmission rates, etc.

  # Initial condition
  S   <- x[1]
  Ihh <- x[2]  # Number of individuals infected by other infected individuals
  Ihw <- x[3]  # Number of individuals infected by water
  Ihf <- x[4]  # Number of individuals infected by food
  Ihc <- x[5]  # Number of individuals infected by live chickens
  Ihe <- x[6]  # Number of individuals infected by soil
  Iho <- x[7]  # Number of individuals infected by other sources
  W   <- x[8]  # Pathogens in water
  F   <- x[9]  # Pathogens in food
  C   <- x[10] # Prevalence among live chickens
  E   <- x[11] # Pathogens in soil
  O   <- x[12] # Pathogens in "other:
  N   <- S + (Ihh + Ihw + Ihf + Ihc + Ihe + Iho)

  # Define each rate moving from one compartment to the other
  gamma  <- parms[7]  # Recovery rate
  alpha  <- parms[8]  # Shedding rate into water and soil
  xi     <- parms[9]  # Clearance rate from water and soil
  mu     <- parms[10] # Birth and death rate

  # Scale transmission rates by gamma+mu
  beta_I <- parms[1]*(gamma+mu)/(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) # Transmission rate from person to
person
  beta_W <- parms[2]*(gamma+mu)/W # Transmission rate from water to person
  beta_F <- parms[3]*(gamma+mu)/F # Transmission rate from food to person
  beta_C <- parms[4]*(gamma+mu)/C # Transmission rate from live chicken to person
  beta_E <- parms[5]*(gamma+mu)/E # Transmission rate from soil to person
  beta_O <- parms[6]*(gamma+mu)/O # Transmission rate from all other sources

  dxdt <- numeric(length(x))
```

```
  dxdt[1] <- mu*N -beta_I*S*(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) - beta_W*S*W - beta_F*S*F - beta_C*S*C -
beta_E*S*E - beta_O*S*O - mu*S + gamma*(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) # dS/dt
  dxdt[2] <-       beta_I*S*(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) - gamma*Ihh - mu*Ihh # dIhh/dt (person-to-
person transmission)
  dxdt[3] <-       beta_W*S*W                         - gamma*Ihw - mu*Ihw # dIhw/dt (water-to-
person transmission)
  dxdt[4] <-       beta_F*S*F                         - gamma*Ihf - mu*Ihf # dIhf/dt (food-to-
person transmission)
  dxdt[5] <-       beta_C*S*C                         - gamma*Ihc - mu*Ihc # dIhc/dt (live
chicken-to-person transmission)
  dxdt[6] <-       beta_E*S*E                         - gamma*Ihe - mu*Ihe # dIhe/dt (soil-to-
person transmission)
  dxdt[7] <-       beta_O*S*O                         - gamma*Iho - mu*Iho # dIho/dt (all other-
to-person transmission)
  dxdt[8] <-       alpha*(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) - xi*W # dW/dt (Pathogen shedding into water
and clearance from water)
  dxdt[9] <- 0 # dF/dt. Constant. Not time varying.
  dxdt[10]<- 0 # dC/dt. Constant. Not time varying.
  dxdt[11]<-       alpha*(Ihh+Ihw+Ihf+Ihc+Ihe+Iho) - xi*E # dE/dt (Pathogen shedding into soil
and clearance from soil)
  dxdt[12]<- 0 # dO/dt. Constant. Not time varying.

  list(dxdt)
}



#-------------------------------------------------------------------------------#
# Sobol sampling for the prevalence and contribution of each pathway
#-------------------------------------------------------------------------------#

# Number of parameter sets you want to run
numsamples <- 2.5e+06

# Number of parameters you want to estiamte
numparam <- 7 # 6 for the Campylobacter model and 7 for the Salmonella model

# Sobol sampling
p <- sobolSequence.points(numparam, count=numsamples)

# Lower and upper bounds for the Latin Hypercubic sampling
lhs_low  <-  rep(0, numparam)
lhs_high <-  c(0.4, rep(1, numparam-1))

# Scale Sobol samples, using the lower and upper bounds defined above
rho <- t(lhs_low + (lhs_high - lhs_low)*t(p)) # Scaled parameter sets

# Need to have these scaled sampled attributable fractions sum to 1.
for (i in 1:numsamples){
  rho[i, 2:numparam] <- rho[i, 2:numparam]/sum(rho[i,2:numparam])
}

#-------------------------------------------------------------------------------#
# Set up parameters
#-------------------------------------------------------------------------------#

# Population size in the simulated population
N <- 1 # Should be 1, as we are modeling the prevalence

# Fixed parameters
gamma <- 1  # Recovery rate
alpha <- 1  # Shedding rate into water
xi    <- 1  # Clearance rate from water (Stays in the environment for 1/xi days)
b     <- 70/1000/365.25 * N
mu    <- 70/1000/365.25      # Background death rate for children <5 yo (70 per 1,000 population
during a year at midyear of 2019)

# Set the initial condition
# (i.e., fraction of individuals and pathogen concentration in each status at Time 1)
S   <- N - 0.1
Ihh <- 0
```

```
Ihw <- 0
Ihf <- 0
Ihc <- 0
Ihe <- 0
Iho <- 0.1
W   <- 0.1 # Pathogen concentration in water, W_t
F   <- 1   # Pathogen concentration in food, F_t is constant over time (set to 1 so that it is
basically included in beta_F)
C   <- 1   # Prevalence of infected live chicken which is constant over time (set to 1 so that it
is basically included in beta_C)
E   <- 0.1 # Pathogen concentration in soil, E_t
O   <- 1   # Pathogen concentration in all other sources is constant over time (set to 1 so that
it is basically included in beta_O)
set.ODEtime <- seq(from=0, to=365.25*2, by=1) # Run the model for two years

# Create a vector containing the initial condition
set.x0        <- c(S[1], Ihh[1], Ihw[1], Ihf[1], Ihc[1], Ihe[1],  Iho[1], W[1], F[1], C[1], E[1],
O[1])
names(set.x0) <- c('S', 'Ihh',  'Ihw',  'Ihf',  'Ihc',  'Ihe',   'Iho',  'W',  'F',  'C',  'E',
'O')


#-------------------------------------------------------------------------------#
# Run ODE with each parameter set and calculate nLL
#-------------------------------------------------------------------------------#

# These are containers that we'll fill in
res.prev  <- matrix(NA, nrow=numsamples, ncol=(numparam)) # estimated prevalence (prev_I,
prev_Ihh, prev_Ihw, prev_Ihf, prev_Ihc, prev_Iho) will be stored here
parsample <- matrix(NA, nrow=numsamples, ncol=(numparam)) # scaled Sobol sampled parameter sets
(prevalence and AFs) will be stored here
nllsample <- numeric(numsamples) # netative log-likelihood will be stored here
env.conc  <- matrix(NA, nrow=numsamples, ncol=2) # pathogen concentration in the environmental
source (water in this case) in the endemic status

# Create a function to run ODE and calculate negative log-likelihood
ode_func <- function (lambda_gm, # prevalence (all pathways) scaled by gamma*mu
                      partition) { # AFs
  out <- ode(y = set.x0,              # initial state values
             t = set.ODEtime,         # time step (t=0,1,2,3,...)
             func   = SIWFCEOS_model, # ODE function
             parms  = c(lambda_gm*partition, gamma, alpha, xi, mu), # Sobol samples and fixed
parameters
             method = 'vode')         # Method that performs integration (lsode, ode45, vode)
  out <- as.data.frame(out)

  # Calculate the proportion of infected children (combining all transmission routes)
  out$I <- out$Ihh + out$Ihw + out$Ihf + out$Ihc + out$Ihe + out$Iho

  # Simulated prevalence at the stable stage (equilibrium)
  prev_Ihh <- tail(out[,"Ihh"],1)
  prev_Ihw <- tail(out[,"Ihw"],1)
  prev_Ihf <- tail(out[,"Ihf"],1)
  prev_Ihc <- tail(out[,"Ihc"],1)
  prev_Ihe <- tail(out[,"Ihe"],1)
  prev_Iho <- tail(out[,"Iho"],1)
  prev_I   <- tail(out[,"I"],  1)
  W_end    <- tail(out[,"W"],  1)
  E_end    <- tail(out[,"E"],  1)

  # Calculate the negative log-likelihood using the MapSan data and the WHO FERG
  # attributable fractions (Multinomial distribution)
  NLL = -sum(759*c(0.21*c(0.18,0.10,0.46,0.15,0.01,0.10), 1-0.21) * log(c(prev_Ihh, prev_Ihw,
prev_Ihf, prev_Ihc, prev_Ihe, prev_Iho, 1-prev_I)))

  # Output
  return(c(NLL, prev_I, prev_Ihh, prev_Ihw, prev_Ihf, prev_Ihc, prev_Ihe, prev_Iho, W_end,
E_end))
}

# Run ODE and calculate negative log-likelihood for each sampled parameter set
```

```
for(i in 1:numsamples){
  if (i%%1000==0) {
    print(i)
  }

  # Scaled Sobol sampled prevalence of infection
  target_prev <- rho[i,1]

  # Scale the prevalence at steady state by (gamma + mu)
  lambda_gm <- target_prev/(1-target_prev)

  # Run ODE
  temp <- ode_func(lambda_gm, # Prevalence scaled by gamma+mu
                   rho[i,-1]) # Scaled Sobol sampled AFs

  # Save outputs
  nllsample[i]  <- temp[1]  # negative log-likelihood
  parsample[i,] <- rho[i,]  # scaled LH sampled parameter sets (prevalence and AFs)
  res.prev[i,]  <- temp[2:(numparam+1)] # estimated prevalence (prev_I, prev_Ihh, prev_Ihw,
prev_Ihf, prev_Ihc, prev_Iho)
  env.conc[i,]  <- temp[c(numparam+2,numparam+3)]
}

parsample <- as.data.frame(parsample)
names(parsample) <- c("baseline.prev", "AF_h", "AF_w", "AF_f", "AF_c", "AF_e", "AF_o")
res.prev <- as.data.frame(res.prev)
names(res.prev) <- c("prev_I", "prev_Ihh", "prev_Ihw", "prev_Ihf", "prev_Ihc", "prev_Ihe",
"prev_Iho")
env.conc <- as.data.frame(env.conc)
names(env.conc) <- c("W_end", "E_end")

#------------------------------------------------------------------------------#
# Resampling
#------------------------------------------------------------------------------#

# Calculate the relative negative log-likelihood
rel.nllsample <- nllsample - min(nllsample)

# Calculate the probability (weights) based on the relative nLL
prob.nllsample <- exp(-rel.nllsample)

# Set the number of samples you want to resample
num.resample <- 10000

# Resample the rows based on the sample importance, prob.nllsample (with replacement)
resample.ind <- sample(1:length(nllsample), num.resample, replace=T, prob=prob.nllsample)
resampled.parsample <- parsample[resample.ind,]
resampled.res.prev  <- res.prev[resample.ind,]
resampled.env.conc  <- env.conc[resample.ind,]
```