APPENDIX A: Data-fitting code

```
#Packages

library('devtools')

library('fitR')

library("coda")

library("lattice")


##data load

Dataseries<-read.table(file.choose(), header=T, na.strings="NA", sep=",")#2011


#Stochastic SEIAR model

SEIAR_name <- "SEIAR Stochastic with constant population size"

SEIAR_state.names <- c("S","E","I", "A","R")

SEIAR_theta.names <- c("epsilon", "b", "mu_h", “gamma”, "lambda", "zeta", "alpha" ,"delta", "pi"," rho"
, “eta”, “mu_v”, “nu”, “psi”, “kappa”,"omga")

SEIAR_stoc <- function(theta,init.state,times) {

  SEIAR_transitions <- list(

    c(S=-1,E=1),# infection

    c(E=-1,I=1, Inc=1), c(E=-1, A=1),  c(I=-1,R=1),

    c(I=-1), c(S=1), c(A=-1,R=1),  c(R=-1,S=1),

    c(S=-1,E=-1, I=-1,A=-1,R=-1),

    c(X=-1,Y=1), c(Y=-1,Z=1),  c(X=-1,Y=-1, Z=-1) )

SEIR_rateFunc <- function(state,theta,t) {


    # Parameters

    Lambda_1<- theta[["epsilon"]]*theta[["b"]]

    Lambda <-  1 / theta[["lambda"]]

    Zeta <- theta[["zeta"]]

 Gamma <- theta[["gamma"]]
```

```
  Alpha <-  1 / theta[[“alpha"]]
  Delta <- theta[["delta"]]
  Pi<- 1 / theta[["pi"]]
  Rho <- 1 / theta[["rho"]]
  mu <- theta[["mu_h"]]
  #Psi <- theta[["psi"]]
  beta_I <- theta[["epsilon"]]*theta[["nu"]]
  beta_A <- theta[["epsilon"]]*theta[["kappa"]]
  eta_v <- 1/theta[["eta"]]
  Mu_v <- theta[["mu_v"]]

  # states
  S <- state[["S"]]
  E <- state[["E"]]
  I <- state[["I"]]
  A <- state[["A"]]
  R <- state[["R"]]
  X <- state[["X"]]
  Y <- state[["Y"]]
  Z <- state[["Z"]]
  Inc <- state[["Inc"]]
  N <- S + E +I+A + R
  return(c( Lambda_1*S*Z/N, Lambda*E, Zeta*E, Alpha *I,   delta *I,
   Gamma, Pi*A, Rho*R,  mu,  beta_I*X*I/N + beta_A*X*A/N, eta_v*Y,  Mu_v ))  }
# put incidence at 0 in init.state
init.state["Inc"] <- 0
traj <- simulateModelStochastic(theta,init.state,times,SEIR_transitions,SEIR_rateFunc)
  #traj$Inc <- c(0, diff(traj$Inc))
return(traj)}
```

```
## Function to compute the likelihood
SEIAR_dpintobs<- function(data.point, model.point, theta, log = FALSE){
  return(dpois(x=data.point[["obs"]],lambda=theta[["omga"]]*model.point[["Inc"]],log=log))  }
## Function to generate observation from a model simulation
SEIR_rpointobs<- function(model.point, theta){
   obs.point <- rpois(n=1, lambda=theta[["omga"]]*model.point[["Inc"]])
  return(c(obs=obs.point))}
simulateModelStochastic<- function (theta, init.state, times, transitions, rateFunc) {
stoch <- as.data.frame(ssa.adaptivetau(init.state, transitions,
             rateFunc, theta, tf = diff(range(times))))
 stoch$time <- stoch$time + min(times)
  traj <- cbind(time = times, apply(stoch[, -1], 2, function(col) {
    approx(x = stoch[, 1], y = col, xout = times, method = "constant")$y
  }))
  return(as.data.frame(traj))}

## create stochastic  SEIARXYZ fitmodel
SEIAR_stoch <- fitmodel(
  name = SEIAR_name,
  state.names = SEIAR_state.names,
  theta.names = SEIR_theta.names,
  simulate = SEIAR_stoc,
  ssa.adaptivetau,
  rPointObs = SEIR_rpointobs,
  dPointObs = SEIR_dpintobs)

#Paramter estimate for WRP
```

theta <- c(epsilon = 32.5, b = 0.84, mu_h = 0.0000541, lambda = 5, Zeta =19,   alpha =18, , delta = 0.00004,   Pi= 220, Rho = 37, kappa=0.4, nu=0.48, mu_v=0.132, eta_v=12, omga= 0.85) #wRP th_alpha include movement

init.state <- c(S = 7983420, E = 0, I = 100, A = 0, R = 0, In=0)

Paramter estimate for WBGZ

theta <- c(epsilon = 29.7, b = 0.84, mu_h = 0.0000541, lambda = 6, Zeta =19,   alpha =20, , delta = 0.00004,   Pi= 190, Rho = 20*365, kappa=0.4, nu=0.48, mu_v=0.132, eta_v=12, omga= 0.85)

init.state <- c(S =9967450 , E = 0, I = 100, A = 0, R = 0, In=00)

Parameter estimate for CES

theta <- c(epsilon = 34.5, b = 0.84, mu_h = 0.0000541, lambda = 5, Zeta =19,   alpha =16, , delta = 0.00004,   Pi= 150, Rho = 25*365, kappa=0.4, nu=0.48, mu_v=0.132, eta_v=12, omga= 0.85)

init.state <- c(S = 8826740, E = 0, I = 240, A = 0, R = 0, In=00) ##end function

Output

trajC <- SEIR_stoch$simulate(theta, init.state, times)

plotTraj(trajC)

plotFit(SEIR_stoch, theta, init.state, data = Dataseries, n.replicates = 1000)#