# AGE STRUCTURED MODELS

AARON A. KING AND HELEN J. WEARING

## Contents

*Date*: June 24, 2011.

## 1. Beyond homogeneous populations: age structure

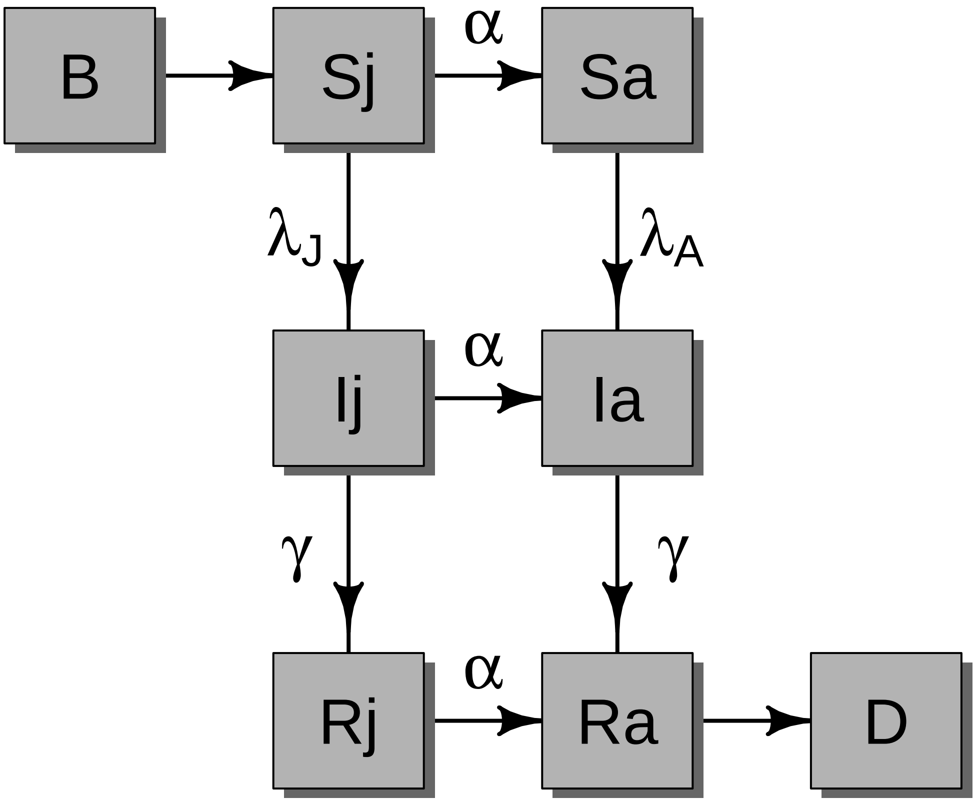

Figure 1. SIR dynamics in an age-structured population.

Today's first lecture showed how force of infection can vary with age. What sort of mechanisms might give rise to these effects? Here we'll see to what extent we can infer these mechanisms on the basis of age-specific incidence and seroprevalence data. We'll start by introducing age into the simplest mechanistic model we can think of, which has separate classes for juveniles and adults:

$$\begin{aligned}
\frac{dS_J}{dt} &= -\lambda_J\,S_J \\
\frac{dS_A}{dt} &= -\lambda_A\,S_A \\
\frac{dI_J}{dt} &= \quad \lambda_J\,S_J - \gamma\,I_J \\
\frac{dI_A}{dt} &= \quad \lambda_A\,S_A - \gamma\,I_A \\
\frac{dR_J}{dt} &= \qquad\qquad + \gamma\,I_J \\
\frac{dR_A}{dt} &= \qquad\qquad + \gamma\,I_A
\end{aligned} \tag{1}$$

The $\lambda$s denote the age-specific force of infections:

$$\begin{aligned}
\lambda_J &= \beta_{JJ}\,I_J + \beta_{JA}\,I_A \\
\lambda_A &= \beta_{AJ}\,I_J + \beta_{AA}\,I_A
\end{aligned} \tag{2}$$

In this model, each population can infect each other but the infection moves through the populations separately. Let's simulate such a model. To make things concrete, we'll assume that the transmission rates $\beta$ are greater within groups than between them.

```
b1 <- 0.005
b2 <- 0.005
gamma <- 10

ja.model <- function (t, x, ...) {
  s <- x[c("Sj","Sa")]                  # susceptibles
  i <- x[c("Ij","Ia")]                  # infecteds
  r <- x[c("Rj","Ra")]                  # recovereds
  n <- s+i+r                            # total pop
  lambda.j <- (b1+b2)*i[1]+b1*i[2]      # juv. force of infection
  lambda.a <- b1*i[1]+(b1+b2)*i[2]      # adult. force of infection
  list(
       c(
         -lambda.j*s[1],
         -lambda.a*s[2],
          lambda.j*s[1]-gamma*i[1],
          lambda.a*s[2]-gamma*i[2],
                        gamma*i[1],
                        gamma*i[2]
         )
       )
}
```

```
require(deSolve)
## initial conditions
yinit <- c(Sj=2000,Sa=1000,Ij=0,Ia=1,Rj=0,Ra=0)
sol <- ode(
           y=yinit,
           times=seq(0,2,by=0.01),
           func=ja.model
           )

plot(sol)
dim(sol)
head(sol)
plot(sol,log='y')

time <- sol[,1]                          # time
y <- sol[,-1]                            # all other variables
n <- apply(y,1,sum)                      # population size
prop <- y/n                              # fractions
subsampled.prop <- prop[seq(1,length(time),by=10),]
subsampled.time <- time[seq(1,length(time),by=10)]
barplot(
        t(subsampled.prop),
        names.arg=subsampled.time,
        xlab='time',main='Population structure',
        space=0,
        col=c(
          rgb(0.5,1,0.5),
          rgb(0,1,0),
          rgb(1,0.5,0.5),
          rgb(1,0,0),
          rgb(0.5,0.5,1),
          rgb(0,0,1)
          ),
        legend = colnames(prop),
        args.legend=list(bg="white")
        )
```

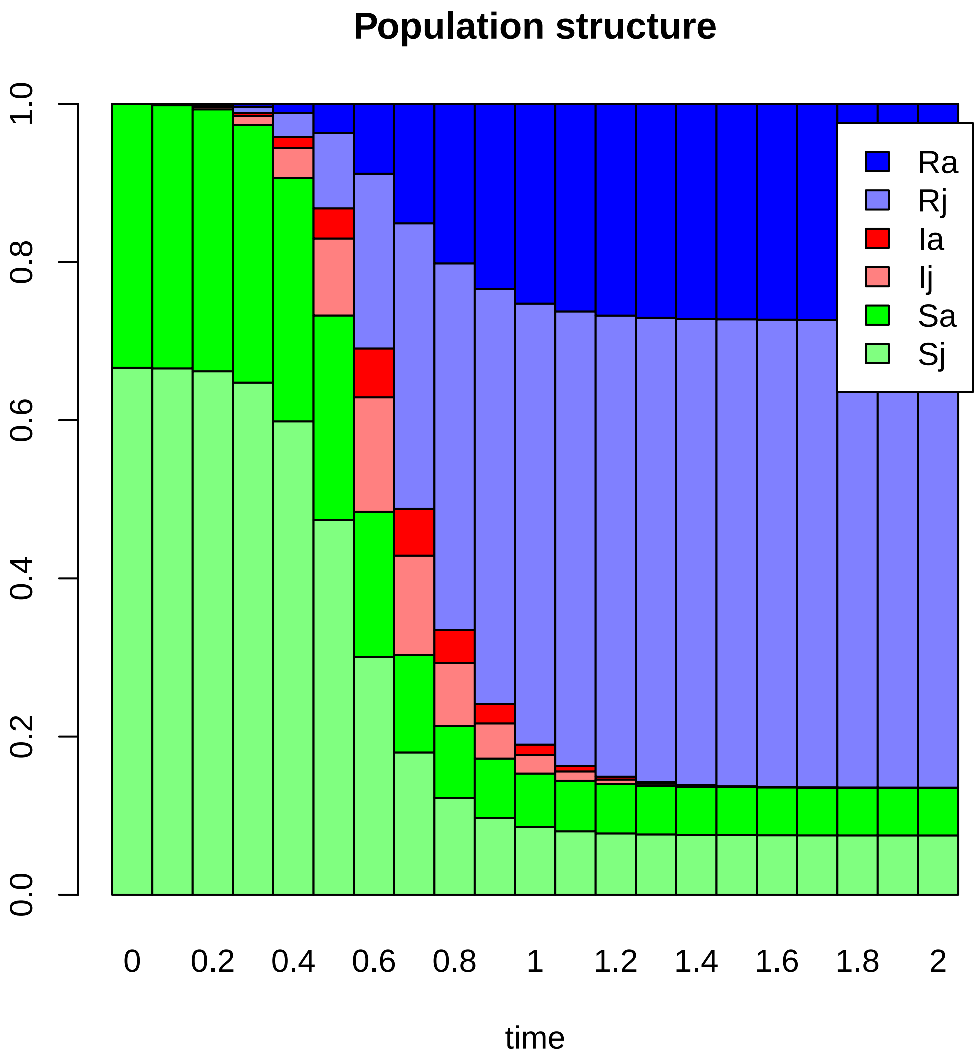

FIGURE 2. The population structure changes over the course of the epidemic.

The results of the above are plotted in Fig. 2. However, we haven't yet modeled the aging process. We can do this very simply using the same ingredients that go into the basic SIR model. In that model, the waiting times in the S and I classes are exponential. Let's assume the same thing about the aging process. We'll also add in births.

$$
\begin{aligned}
\frac{dS_J}{dt} &= -\lambda_J\,S_J && + B && - \alpha\,S_J \\
\frac{dS_A}{dt} &= -\lambda_A\,S_A && && - \mu\,S_A + \alpha\,S_J \\
\frac{dI_J}{dt} &= \lambda_J\,S_J - \gamma\,I_J && && - \alpha\,I_J \\
\frac{dI_A}{dt} &= \lambda_A\,S_A - \gamma\,I_A && && - \mu\,I_A + \alpha\,I_J \\
\frac{dR_J}{dt} &= + \gamma\,I_J && && - \alpha\,R_J \\
\frac{dR_A}{dt} &= + \gamma\,I_A && && - \mu\,R_A + \alpha\,R_J
\end{aligned}
\tag{3}
$$

Now, let's simulate this model, under the same assumptions about transmission rates as above.

```
b1 <- 0.005
b2 <- 0.005
gamma <- 10
births <- 100
da <- c(20,60)                          # alpha = 1/da

ja.demog.model <- function (t, x, ...) {
  s <- x[c("Sj","Sa")]                  # susceptibles
  i <- x[c("Ij","Ia")]                  # infecteds
  r <- x[c("Rj","Ra")]                  # recovereds
  n <- s+i+r                            # total pop
  lambda.j <- (b1+b2)*i[1]+b1*i[2]      # juv. force of infection
  lambda.a <- b1*i[1]+(b1+b2)*i[2]      # adult. force of infection
  alpha <- 1/da
  list(
       c(
         -lambda.j*s[1]           -alpha[1]*s[1]+births,
         -lambda.a*s[2]           +alpha[1]*s[1]-alpha[2]*s[2],
          lambda.j*s[1]-gamma*i[1]-alpha[1]*i[1],
          lambda.a*s[2]-gamma*i[2]+alpha[1]*i[1]-alpha[2]*i[2],
                        gamma*i[1]-alpha[1]*r[1],
                        gamma*i[2]+alpha[1]*r[1]-alpha[2]*r[2]
         )
       )
}
```

Note that in this function, $\mu =$`alpha[2]`, i.e., death is just another age class.

```
require(deSolve)
## initial conditions
yinit <- c(Sj=2000,Sa=1000,Ij=0,Ia=1,Rj=0,Ra=5000)
sol <- ode(
           y=yinit,
           times=seq(0,200,by=0.1),
           func=ja.demog.model
           )
```

```
plot(sol)
dim(sol)
head(sol)
plot(sol,log='y')

equil <- drop(tail(sol,1))[-1]
n <- equil[c("Sj","Sa")]+equil[c("Ij","Ia")]+equil[c("Rj","Ra")]
seroprev <- equil[c("Rj","Ra")]/n
names(seroprev) <- c("J","A")
barplot(height=seroprev,width=da,ylab="seroprevalence")
```

To compute $R_0$, we need to know the stable age distribution of the population, which we can find by solving for the disease-free equilibrium: $S_J^* = B/\alpha$ and $S_A^* = B/\mu$.

Now we have the stable age distribution, we can calculate $R_0$ by constructing the next generation matrix. Intuitively, this is a matrix that specifies how many new age-specific infections are generated by a typical infected individual of each age class (in a fully susceptible population). For example, let's consider an infected adult and ask how many new juvenile infections it generates: this is the product of the number of susceptible juveniles (from the stable age distribution), the per capita transmission rate from adults to juveniles and the average duration of infection, i.e. $S_J^* \times \beta_{JA} \times 1/(\gamma+\mu)$. This forms one element of our next generation matrix. The other elements look very similar, except there are extra terms when we consider an infected juvenile because there is a (very small) chance they may age during the infectious period and therefore cause new infections as an adult:

$$\mathrm{NGM} = \begin{pmatrix} \frac{S_J^*\beta_{JJ}}{(\gamma+\alpha)} + \frac{\alpha}{(\gamma+\mu)}\frac{S_J^*\beta_{JA}}{(\gamma+\mu)} & \frac{S_J^*\beta_{JA}}{(\gamma+\mu)} \\ \frac{S_A^*\beta_{AJ}}{(\gamma+\alpha)} + \frac{\alpha}{(\gamma+\mu)}\frac{S_A^*\beta_{AA}}{(\gamma+\mu)} & \frac{S_A^*\beta_{AA}}{(\gamma+\mu)} \end{pmatrix} \tag{4}$$

$R_0$ can then be computed as the dominant eigenvalue (i.e., the one with the largest real part) of this matrix. In R, we do

```
alpha <- 1/da[1]
mu <- 1/da[2]
n <- births/c(alpha,mu)
beta <- matrix(c(b1+b2,b1,b1,b1+b2),nrow=2,ncol=2)
ngm <- matrix(
              c(
                n[1]*beta[1,1]/(gamma+alpha)+alpha/(gamma+mu)*n[1]*beta[1,2]/(gamma+mu),
                n[2]*beta[2,1]/(gamma+alpha)+alpha/(gamma+mu)*n[2]*beta[2,2]/(gamma+mu),
                n[1]*beta[1,2]/(gamma+mu),
                n[2]*beta[2,2]/(gamma+mu)
                ),
              nrow=2,
              ncol=2
              )
eigen(ngm)
eigen(ngm,only.values=TRUE)
max(Re(eigen(ngm,only.values=T)$values))
```

## 2. Getting more realistic: nobody's youth is exponential

In the model above, the aging process follows an exponential distribution, which means that whether an individual is 1 year old or 10 years old, the chance of them becoming an adult is the same! To improve on this, we can assume that the time a juvenile must wait before becoming an adult follows a gamma distribution. This is equivalent to saying that the waiting time is a sum of some number of exponential distributions. This suggests that we can achieve such a distribution by adding age classes to the model, so that becoming an adult means passing through some number of stages. We'll use 30 age classes, and since they don't have to be of equal duration, we'll assume that they're not. Specifically, we'll have 20 1-yr age classes to take us up to adulthood and break adults into 10 age classes of 5 yr duration each. The last one will take us up through age 80.

Now, when we had just two age classes, we could write out each of the equations easily enough, but now that we're going to have 30, we'll need to be more systematic. In particular, we'll need to think of $\beta$ as a matrix of transmission rates. Let's see how to define such a matrix in R. So that we don't change too many things all at once, let's keep the same contact structure as in the juvenile-adult model.

```
ages <- c(seq(1,20,by=1),seq(25,65,by=5),80) # upper end of age classes
da <- diff(c(0,ages))                        # widths of age classes
beta <- matrix(nrow=30,ncol=30)
beta[1:20,1:20] <- b1+b2
beta[21:30,21:30] <- b1+b2
beta[1:20,21:30] <- b1
beta[21:30,1:20] <- b1
dim(beta)
filled.contour(beta,plot.title=title(main="WAIFW matrix"))
```

Let's use the techniques we learned from John to simulate the re-introduction of a pathogen into a population of hosts. We'll assume that, at the time of introduction, all children are susceptible, as are adults over 45, but that individuals aged 20–45 have seen the pathogen before and are immune. The vector `yinit` expresses these initial conditions.

```
yinit <- c(
           S=c(rep(100,20),rep(0,5),rep(200,5)),
           I=c(rep(0,25),1,rep(0,4)),
           R=c(rep(0,20),rep(1000,5),rep(0,5))
           )
```

The codes that follow will be a bit easier to follow if we introduce some indexes that will allow us to pick out certain bits of the `yinit` vector.

```
sindex <- 1:30
iindex <- 31:60
rindex <- 61:90
juvies <- 1:20
adults <- 21:30
```

Now, to capture the aging process, it's convenient to define another matrix to hold the rates of movement between age classes.

```
aging <- diag(-1/da)
aging[row(aging)-col(aging)==1] <- 1/head(da,-1)
```

Have a look at the `aging` matrix, for example by doing:

```
aging[1:5,1:5]
aging[1:5,6:10]
aging[25:30,25:30]
filled.contour(aging)
```

What can you say about its structure?

Now we can put the pieces together to write a simulator for the age-structured SIR dynamics.

```
ja.multistage.model <- function (t, x, ...) {
  s <- x[sindex]                  # susceptibles
  i <- x[iindex]                  # infecteds
  r <- x[rindex]                  # recovereds
  lambda <- beta%*%i              # force of infection
  dsdt <- -lambda*s+aging%*%s
  didt <-  lambda*s+aging%*%i-gamma*i
  drdt <-           aging%*%r+gamma*i
  dsdt[1] <- dsdt[1]+births
  list(
       c(
         dsdt,
         didt,
         drdt
         )
       )

}
```

We can plug this into `ode` just as we did the simpler models to simulate an epidemic. We'll then plot the epidemic curve.

```
sol <- ode(
           y=yinit,
           times=seq(0,100,by=0.1),
           func=ja.multistage.model
           )
time <- sol[,1]
infects <- sol[,1+iindex]
plot(time,apply(infects,1,sum),type='l')
lines(time,apply(infects[,juvies],1,sum),col='red')
lines(time,apply(infects[,adults],1,sum),col='blue')
```

Let's mimic a situation where we have cross-sectional seroprevalence data. In using such data, we'd typically assume that the system was at equilibrium. What does the equilibrium age-specific seroprevalence look like in this example?

```
equil <- drop(tail(sol,1))[-1]
n <- equil[sindex]+equil[iindex]+equil[rindex]
seroprev <- equil[rindex]/n
names(seroprev) <- ages
barplot(height=seroprev,width=da)
```

Let's also compute $R_0$. To do so, we'll need the stable age distribution. We can get that by simulating an infection-free population:

```
yinit.sonly <- c(
                 S=c(rep(250,30)),
                 I=c(rep(0,30)),
                 R=c(rep(0,30))
                 )
sol <- ode(
           y=yinit.sonly,
           times=seq(0,300,by=1),
           func=ja.multistage.model
           )
time <- sol[,1]
pop <- apply(sol[,-1],1,sum)
plot(time,pop,type='l')
```

Alternatively, we can get the stable age distribution by finding the population structure that balances the birth, aging, and death processes. At equilibrium, we have the matrix equation

$$\begin{pmatrix} -\alpha_1 & 0 & 0 & \cdots & 0 \\ \alpha_1 & -\alpha_2 & 0 & \cdots & 0 \\ 0 & \alpha_2 & -\alpha_3 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & & \alpha_{29} & -\alpha_{30} \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_{30} \end{pmatrix} + \begin{pmatrix} B \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

To solve this equation in R, we can do

```
## get stable age distribution
n <- solve(aging,-c(births,rep(0,29)))
```

The following lines then compute $R_0$. This calculation comes from a recipe described in detail by Diekmann and Heesterbeek (2000) and Hurford *et al.* (2010).

```
F <- diag(n)%*%beta+aging-diag(diag(aging))
V <- diag(gamma-diag(aging))
max(Re(eigen(solve(V,F),only.values=T)$values))
```

**Exercise 1.** Modify the structure of the transmission matrix to reflect reasonable assumptions about the mixing of different age-groups in human populations. Use `image` or `filled.contour` to plot the $\beta$ matrix. Compute $R_0$ for your assumptions. Simulate and plot the age-structured SIR dynamics under your assumptions and determine the equilibrium prevalence and seroprevalence.

## 3. What do real contact networks look like?

The POLYMOD study (Mossong *et al.*, 2008) was a journal-based look into the contact network in contemporary European society. Let's have a look what these data tell us about the contact structure.

```
moss <- read.csv(
                 url("http://www.math.mcmaster.ca/~bolker/eeid/data/mossong.csv"),
                 as.is=TRUE
                 )
age.categories <- moss$contactor[1:30]
moss$contactor <- ordered(moss$contactor,levels=age.categories)
moss$contactee <- ordered(moss$contactee,levels=age.categories)
```

Since contacts are symmetric, we'll need to estimate the symmetric contact matrix.

```
x1 <- with(
           moss,
           tapply(contact.rate,list(contactor,contactee),unique)
           )
xsym <- (x1+t(x1))/2

filled.contour(ages,ages,log10(xsym))
filled.contour(
               ages,ages,log10(xsym),
               plot.title=title(
                 main=quote(log[10](contact~rate)),
                 xlab="age",ylab="age")
               )
barplot(height=apply(x1,1,sum))
barplot(height=apply(x1,2,sum))
```

While this matrix tells us how many contacts are made per year by an individual of each age, it doesn't tell us anything about the probability that a contact results in communication of infection. Let's assume that each contact has a constant probability $q$ of resulting in a transmission event.

```
q <- 3e-5
beta <- q*xsym
filled.contour(ages,ages,log10(beta))
```

Now let's simulate the introduction of such a pathogen into a population characterized by this contact structure.

```
sol <- ode(
           y=yinit,
           times=seq(0,200,by=0.5),
           func=ja.multistage.model
           )
time <- sol[,1]
infects <- sol[,1+iindex]
plot(time,apply(infects,1,sum),type='l')
lines(time,apply(infects[,juvies],1,sum),col='red')
lines(time,apply(infects[,adults],1,sum),col='blue')
```

As before, we can also look at the equilibrium seroprevalence

```
equil <- drop(tail(sol,1))[-1]
n <- equil[sindex]+equil[iindex]+equil[rindex]
seroprev <- equil[rindex]/n
names(seroprev) <- ages
barplot(height=seroprev,width=da)
```

and compute the $R_0$ for this infection.

```
n <- solve(aging,-c(births,rep(0,29)))
F <- diag(n)%*%beta+aging-diag(diag(aging))
V <- diag(gamma-diag(aging))
max(Re(eigen(solve(V,F),only.values=T)$values))
```

**Exercise 2.** Do

```
sero <- read.csv(url("http://www.math.mcmaster.ca/~bolker/eeid/data/seroprev.csv"))
```

to get some simulated age-specific seroprevalence data. Estimate the force of infection for each of these series using the catalytic model. Discuss your findings. What can you conclude about the contact structure in the populations from which these data are generated?

## References

O. Diekmann and J. A. P. Heesterbeek. *Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation* (J. Wiley and Sons, Chichester, 2000).

A. Hurford, D. Cownden, and T. Day. Next-generation tools for evolutionary invasion analyses. *Journal of the Royal Society Interface* **7**:561–571 (2010).

J. Mossong, N. Hens, M. Jit, P. Beutels, K. Auranen, R. Mikolajczyk, M. Massari, S. Salmaso, G. S. Tomba, J. Wallinga, J. Heijne, M. Sadkowska-Todys, M. Rosinska, and W. J. Edmunds. Social contacts and mixing patterns relevant to the spread of infectious diseases. *PLoS Medicine* **5**:e74 (2008).

A. A. King, Departments of Ecology & Evolutionary Biology and Mathematics, University of Michigan, Ann Arbor, Michigan 48109-1048 USA

*E-mail address*: kingaa at umich dot edu

H. J. Wearing, Departments of Biology and Mathematics & Statistics, University of New Mexico, Albuquerque, NM 87131-0001 USA

*E-mail address*: hwearing at unm dot edu