DAY 1:
INDIVIDUAL MODELS

NME WORKSHOP

Martina Morris, Ph.D.
Steven M. Goodreau, Ph.D.
Samuel M. Jenness, Ph.D.

Supported by the US National Institutes of Health

Model frameworks

In the epidemic modeling literature, we often see two poles:

- Deterministic compartmental models
- Stochastic network models

But there a continuum between these two poles.

Let us tease these apart by considering

- Deterministic compartmental models (done)
- Stochastic compartmental models (brief discussion)
- Stochastic individual-based models (poker-chips, what you just saw, this session)
- Stochastic network models (rest of the week)

Deterministic vs. stochastic models

- To begin: how does stochasticity change a compartmental model?
  - Always via the transitions rate parameters or "flows" in the model

- Consider a simple proportional growth model
  - States: only I is tracked; population has an infinite number of susceptibles
  - Transition rate parameters: only β, the average growth rate of infection

- As a compartmental model, this would be:

$$i(t+1) = i(t) + \beta i(t)$$

which we can also write as:

$$incidence(t) = \beta i(t) \qquad \text{where: } incidence(t) = i(t+1) - i(t)$$

Deterministic vs. stochastic models

	Deterministic	Stochastic
Incidence	$incidence(t) = \beta i(t)$	$P(incidence(t) = k \mid \beta, i(t))$
Incidence is	determined by a rate β	drawn from a probability distribution that depends on β

Deterministic vs. stochastic models

- What does $P(incidence(t) = k \mid \beta, i(t))$ equal?
  - Depends on the model you choose for P(●)
  - P(●) is a probability distribution
  - Probability of what? That the count of new infections = k at time *t*
- So what kind of distributions are appropriate?
  - proper probability distributions (sum across *k* must equal 1)
  - that are discrete
  - on non-negative integers only
- Can you think of an appropriate distribution here?

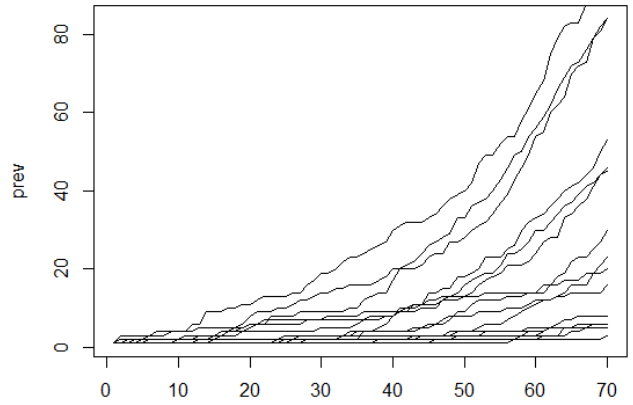
Deterministic vs. stochastic models

- the Poisson distribution
  - Used to model the number of events per time unit (or spatial unit)
  - Arises when the events occur independently with equal probability
  - Defined by one parameter, λ , which is both the mean and variance
  - Range: 0, 1, 2, … (the non-negative integers)
  - The *probability mass function* (PMF) is given by:

$$P(X = k) = \frac{\lambda\, e^{-\lambda}}{k!} \qquad \text{denoted: } Poisson(\lambda)$$

  - What would it look like if we drew the incident infections at each time step from a Poisson distribution with mean $\lambda = \beta i(t)$?

Deterministic vs. stochastic models



Each line represents the epidemic trajectory under a different value of β,

with $incidence(t)$ a stochastic draw from a Poisson($\lambda = \beta i(t)$) distribution

- This approach of drawing the incidence from a distribution at each time step is one way to add stochasticity to a compartmental model
  - Provides a means to quantify the potential variation in outcomes
- But note that we are still only counting aggregates – there are no explicitly represented individuals
  - This limits our ability to represent the relevant aspects of individual heterogeneity

Individual models

- Individual elements are represented explicitly in this model
  - poker chips, EpiModel lab
- Mechanics are different from the very outset
  - While the two approaches may look similar for a simple SI model, the divergence quickly grows as more as more complexity is added
  - Individual models allow for much more fine-grained heterogeneity, and many more forms of it, in a tractable way

Individual SIR model: basic structure

```
# create individuals
# assign status

for (at=1:num.timesteps) {
     # infection
         # calculate number of acts
         # determine who has an act with whom (ie construct "edgelist")
         # limit edgelist to discordant pairs
         # determine infections
         # do bookkeeping for infections
     # recovery
         # identify infecteds
         # determine recoveries
         # do bookkeeping for recoveries
}

# process output
```

Individual models: setup

- create individuals

```
ids <- 1:num                          # num = the initial # of individuals
```

- assign status

```
status <- rep("s", num)               # init.inum = the initial # of infecteds
status[sample(ids,
          size = init.inum)] <- "i"
```

```
> status
[1] "s" "s" "s" "i" "s" "s" "s" "i" "i" "i" "i" "s" "s" "i" "s" "s" "s"
```

Individual models: Infection process

- Step 1: calculate number of acts

```
# n Acts per Time Step = fixed act rate * n/2
acts <- round(act.rate * num[at - 1] / 2)
```

Individual models : Infection process

- Step 2: determine who has an act with whom

```
# Make edgelist of partnerships by ID number
el <- t(replicate(acts, sample(1:num, 2)))
```

```
     [,1] [,2]
[1,]   80    9
[2,]    9   59
[3,]    5   66
[4,]    4   84
```

Individual models: Infection process

- Step 3: limit edge list to discordant pairs

```
# look up the status of each member of the pair
discordant <- (status[el[, 1]] == "i" & status[el[, 2]] == "s") |
         (status[el[, 1]] == "s" & status[el[, 2]] == "i")
[1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE

# create a "discordant edgelist"
del <- el[discordant == TRUE, ]
     [,1] [,2]
[1,]   80    9
[2,]    9   59
[3,]    5   66
[4,]   29   38
```

Individual models: Infection process

- Step 4: determine infections

```
# Infection is a Bernoulli draw for each discordant pair
infections <- rbinom(nrow(del), 1, tprob)

> infections
[1] 1 0 0 1
```

Individual models: Infection process

- Step 5: bookkeeping for infections

```
# Limit discordant edge list to pairs with incident infection
del <- del[infections == TRUE, ]

# Look up newly infected ID in each pair
susIds <- ifelse(status[del[, 1]] == "s", del[, 1], del[, 2])
newInfIds <- susIds[infections == 1]

# Update individual-level status attribute
status[newInfIds] <- "i"
```

Individual models: Recovery process

```
# Identify infected (persons eligible to recover)
idsElig <- which(active == 1 & status == "i")
nElig <- length(idsElig)

# Draw random numbers to determine recoveries
vecRecov <- which(rbinom(nElig, 1, rec.rate) == 1)

# Do bookkeeping
if (length(vecRecov) > 0) {
    idsRecov <- idsElig[vecRecov]
    nRecov <- length(idsRecov)
    status[idsRecov] <- "r"
}
```

Individual models

- Process output

```
# Calculate summary statistics
prevalence <- sum(status == "i")
incidence <- length(newInfIds)
```

Individual SIR model: basic structure

```
# create individuals
# assign status

for (at=1:num.timesteps) {
     # infection
         # calculate number of acts
         # determine who has an act with whom (ie construct "edgelist")
         # limit edgelist to discordant pairs
         # determine infections
         # do bookkeeping for infections
     # recovery
         # identify infecteds
         # determine recoveries
         # do bookkeeping for recoveries
}

# process output
```

Code summary

```
ids <- 1:num                                                    # initial # of individuals
status <- rep("s", num)
status[sample(ids,  size = init.inum)] <- "i"                   # initial # of infecteds

acts <- round(act.rate * num[at - 1] / 2)                       # n Acts per Time Step
el <- t(replicate(acts, sample(1:num, 2)))                     # Edgelist of partnerships by ID
discordant <- (status[el[, 1]] == "i" & status[el[, 2]] == "s") |
         (status[el[, 1]] == "s" & status[el[, 2]] == "i")      # Status lookup
del <- el[discordant == TRUE, ]                                 # Find "discordant edgelist"
infections <- rbinom(nrow(del), 1, tprob)                       # Infection is a Bernoulli draw
del <- del[infections == TRUE, ]                                # Incident pairs
susIds <- ifelse(status[del[, 1]] == "s", del[, 1], del[, 2])   # Inci ID lookup
newInfIds <- susIds[infections == 1]                            # Update individual infection status
status[newInfIds] <- "i"
idsElig <- which(active == 1 & status == "i")
nElig <- length(idsElig)

vecRecov <- which(rbinom(nElig, 1, rec.rate) == 1)              # Recovery is a Bernoulli draw
if (length(vecRecov) > 0) {                                     # Update individual recovery status
    idsRecov <- idsElig[vecRecov]
    nRecov <- length(idsRecov)
    status[idsRecov] <- "r"
}
prevalence <- sum(status == "i")                                # Calculate summary statistics
incidence <- length(newInfIds)
```

Individual models: *Revisiting partnerships*

- Step 2: determine who has an act with whom

```
# Make edgelist of partnerships by ID number
el <- t(replicate(acts, sample(1:num, 2)))
```

```
     [,1] [,2]
[1,]   80    9
[2,]    9   59
[3,]    5   66
[4,]    4   84
```

Individual models

- Was there relational persistence (multiple acts with the same person)?
  - No. How would you add this?
- Were there an constraints on the # of partners per person per time step?
  - No. How would you add this?
- How did we decide who had an act with whom?
  - Purely randomly.
- What if you wanted to add more structure to that process here, in a way that matched observed data. E.g.:
  - Assortative mixing
  - Differential partnership formation rates by attribute or by individual
  - A monogamy bias
  - A tendency to form triangles
  - ***All of these at the same time*** ***NETWORK MODELS!***

Session summary

- One can add stochastic transitions to compartmental models
  - This buys you some quantification of the uncertainty in outcome for that model, but not much else
- Individual models allow one to better represent heterogeneities among elements/actors in the model
  - But control over their relational structure is quite limited
- Network models are needed to explicitly represent systematic patterns in pair formation and dissolution
  - And it is important that these tools be data driven, hence statistical.