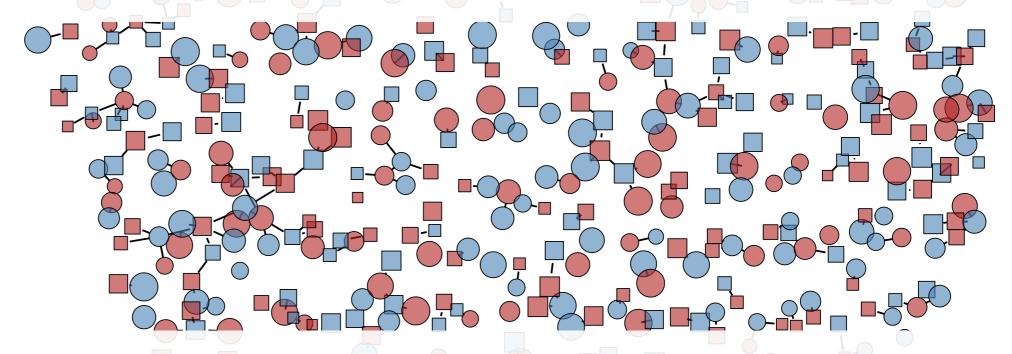
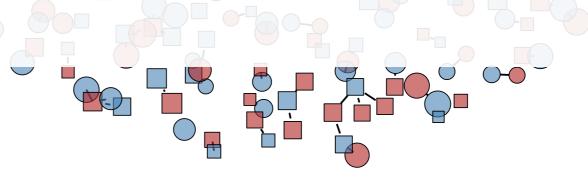
# EpiModel Overview



### **Network Modeling for Epidemics**

Day 3



## Outline for Next 3 Days

### Wednesday

- Modeling epidemics + networks = modeling epidemics over networks
- Core assumption: independent simulations = closed populations
  - Still, network structure ⇒ epidemiology
- Built-in epidemiology from EpiModel

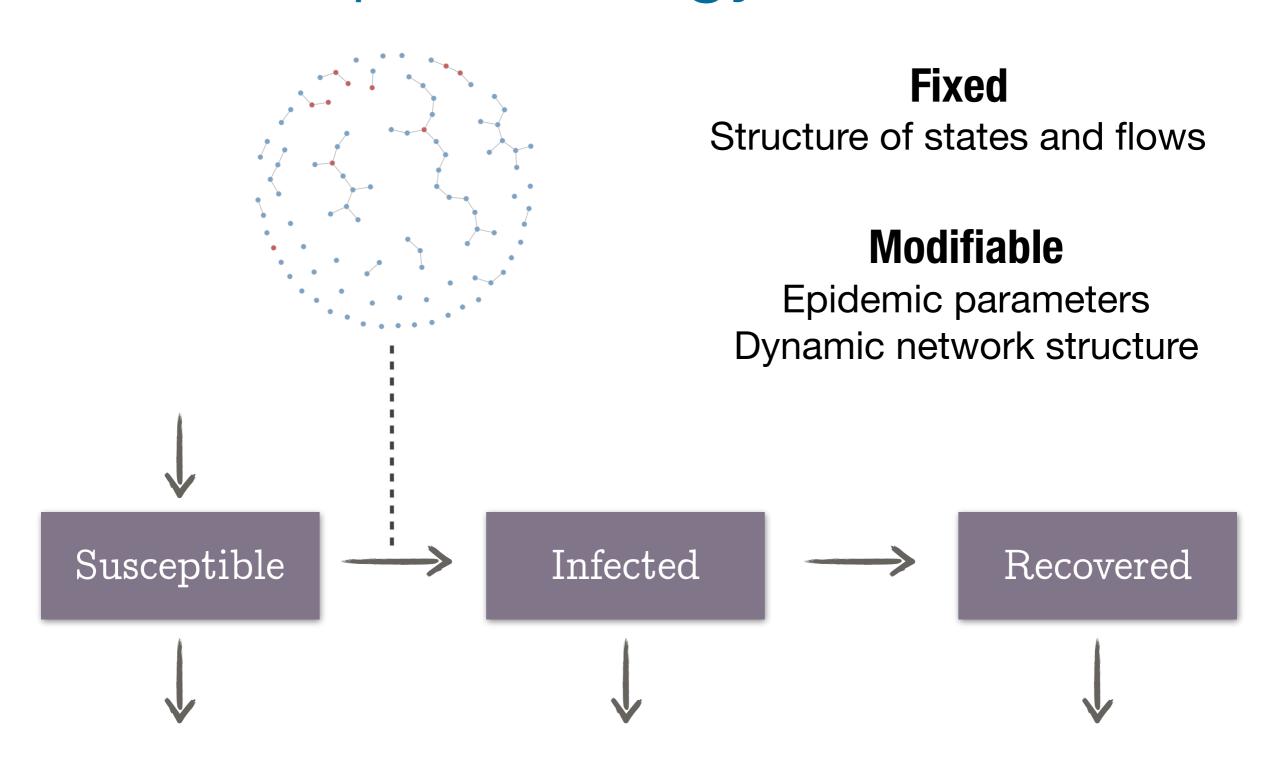
### Thursday

- Dependence: epidemiology (+ everything else) ⇒ network structure
- Vital dynamics, "sero-sorting"
- Built-in epidemiology from EpiModel

### Friday

Extending EpiModel for your research questions

# "Built-in Epidemiology"



```
sti recov <- function(dat, at) {</pre>
                                                                                 recovRGC <- c(recovRGC_asympt, recovRGC_tx, recovRGC_ntx)</pre>
                                                                                 recovUGC <- c(recovUGC asympt, recovUGC tx, recovUGC ntx)</pre>
# Parameters
rgc.dur.asympt <- dat$param$rgc.dur.asympt</pre>
                                                                                dat$attr$rGC[recovRGC] <- 0</pre>
ugc.dur.asympt <- dat$param$ugc.dur.asympt</pre>
                                                                                dat$attr$rGC.sympt[recovRGC] <- NA</pre>
gc.dur.tx <- dat$param$gc.dur.tx</pre>
                                                                                dat$attr$rGC.infTime[recovRGC] <- NA</pre>
gc.dur.ntx <- dat$param$gc.dur.ntx</pre>
                                                                                dat$attr$rGC.tx[recovRGC] <- NA</pre>
rct.dur.asympt <- dat$param$rct.dur.asympt</pre>
                                                                                dat$attr$uGC[recovUGC] <- 0</pre>
                                                                                dat$attr$uGC.sympt[recovUGC] <- NA</pre>
uct.dur.asympt <- dat$param$uct.dur.asympt</pre>
                                                                                dat$attr$uGC.infTime[recovUGC] <- NA</pre>
ct.dur.tx <- dat$param$ct.dur.tx</pre>
ct.dur.ntx <- dat$param$ct.dur.ntx</pre>
                                                                                dat$attr$uGC.tx[recovUGC] <- NA</pre>
                                                                                 dat$attr$GC.cease[c(recovRGC, recovUGC)] <- NA</pre>
# GC recovery
idsRGC asympt <- which(dat$attr$rGC == 1 & dat$attr$rGC.infTime < at &
                                                                                # CT recovery
                            dat$attr$rGC.sympt == 0)
                                                                                idsRCT asympt <- which(dat$attr$rCT == 1 & dat$attr$rCT.infTime < at &</pre>
idsUGC asympt <- which(dat$attr$uGC == 1 & dat$attr$uGC.infTime < at &</pre>
                                                                                                           dat$attr$rCT.sympt == 0)
                            dat$attr$uGC.sympt == 0)
                                                                                idsUCT asympt <- which(dat$attr$uCT == 1 & dat$attr$uCT.infTime < at &</pre>
idsRGC tx <- which(dat$attr$rGC == 1 & dat$attr$rGC.infTime < at &</pre>
                                                                                                           dat$attr$uCT.sympt == 0)
                       dat$attr$rGC.sympt == 1 & dat$attr$rGC.tx == 1)
                                                                                idsRCT tx <- which(dat$attr$rCT == 1 & dat$attr$rCT.infTime < at &</pre>
idsUGC tx <- which(dat$attr$uGC == 1 & dat$attr$uGC.infTime < at &</pre>
                                                                                                       dat$attr$rCT.sympt == 1 & dat$attr$rCT.tx == 1)
                       dat$attr$uGC.sympt == 1 & dat$attr$uGC.tx == 1)
                                                                                idsUCT tx <- which(dat$attr$uCT == 1 & dat$attr$uCT.infTime < at &</pre>
idsRGC ntx <- which(dat$attr$rGC == 1 & dat$attr$rGC.infTime < at &
                                                                                                       dat$attr$uCT.sympt == 1 & dat$attr$uCT.tx == 1)
                        dat$attr$rGC.sympt == 1 & dat$attr$rGC.tx == 0)
                                                                                idsRCT ntx <- which(dat$attr$rCT == 1 & dat$attr$rCT.infTime < at &</pre>
idsUGC ntx <- which(dat$attr$uGC == 1 & dat$attr$uGC.infTime < at &</pre>
                                                                                                        dat$attr$rCT.sympt == 1 & dat$attr$rCT.tx == 0)
                        dat$attr$uGC.sympt == 1 & dat$attr$uGC.tx == 0)
                                                                                 idsUCT ntx <- which(dat$attr$uCT == 1 & dat$attr$uCT.infTime < at &
                                                                                                        dat$attr$uCT.sympt == 1 & dat$attr$uCT.tx == 0)
recovRGC asympt <- idsRGC asympt[which(rbinom(length(idsRGC asympt), 1,</pre>
                                                  1/rgc.dur.asympt) == 1)]
                                                                                 recovRCT asympt <- idsRCT asympt[which(rbinom(length(idsRCT asympt),</pre>
                                                                                                                                  1, 1/rct.dur.asympt) == 1)]
recovUGC_asympt <- idsUGC_asympt[which(rbinom(length(idsUGC_asympt), 1,</pre>
                                                                                 recovUCT asympt <- idsUCT asympt[which(rbinom(length(idsUCT asympt),</pre>
                                                  1/ugc.dur.asympt) == 1)]
                                                                                                                                  1, 1/uct.dur.asympt) == 1)]
recovRGC tx <- idsRGC tx[which(rbinom(length(idsRGC tx), 1,</pre>
                                          1/gc.dur.tx) == 1)
                                                                                 recovRCT tx <- idsRCT tx[which(rbinom(length(idsRCT tx),</pre>
recovUGC tx <- idsUGC tx[which(rbinom(length(idsUGC tx), 1,</pre>
                                                                                                                          1, 1/ct.dur.tx) == 1)]
                                          1/gc.dur.tx) == 1)
                                                                                 recovUCT tx <- idsUCT tx[which(rbinom(length(idsUCT tx),</pre>
                                                                                                                          1, 1/ct.dur.tx) == 1)]
if (!is.null(gc.dur.ntx)) {
   recovRGC ntx <- idsRGC_ntx[which(rbinom(length(idsRGC_ntx), 1,</pre>
                                                                                if (!is.null(ct.dur.ntx)) {
                                              1/gc.dur.ntx) == 1)]
                                                                                   recovRCT ntx <- idsRCT ntx[which(rbinom(length(idsRCT ntx),</pre>
   recovUGC ntx <- idsUGC ntx[which(rbinom(length(idsUGC ntx), 1,</pre>
                                                                                                                              1, 1/ct.dur.ntx) == 1)
                                              1/gc.dur.ntx) == 1)
                                                                                  recovUCT ntx <- idsUCT ntx[which(rbinom(length(idsUCT ntx),</pre>
} else {
                                                                                                                              1, 1/ct.dur.ntx) == 1)]
   recovRGC ntx <- idsRGC ntx[which(rbinom(length(idsRGC ntx), 1,</pre>
                                                                                } else {
                                              1/rgc.dur.asympt) == 1)]
                                                                                   recovRCT ntx <- idsRCT ntx[which(rbinom(length(idsRCT ntx),</pre>
  recovUGC ntx <- idsUGC ntx[which(rbinom(length(idsUGC ntx), 1,</pre>
                                                                                                                              1, 1/rct.dur.asympt) == 1)]
                                              1/ugc.dur.asympt) == 1)
                                                                                   recovUCT ntx <- idsUCT ntx[which(rbinom(length(idsUCT ntx),</pre>
                                                                                                                              1, 1/uct.dur.asympt) == 1)]
```

## Outline for Next 3 Days

### Wednesday

- Modeling epidemics + networks = modeling epidemics over networks
- Core assumption: independent simulations = closed populations
  - Still, network structure ⇒ epidemiology
- Built-in epidemiology from EpiModel

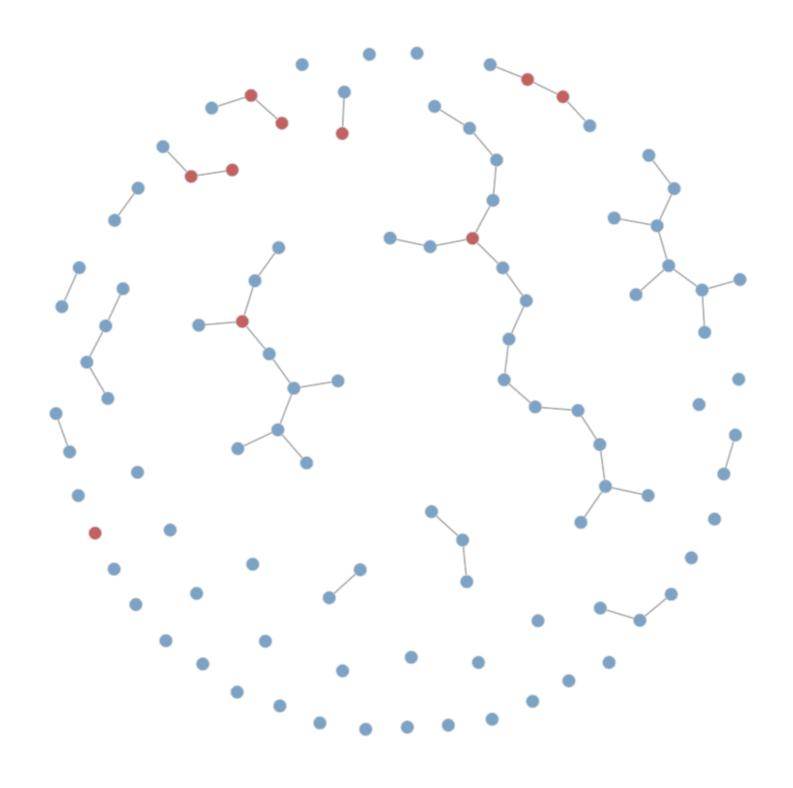
### Thursday

- Dependence: epidemiology (+ everything else) ⇒ network structure
- Vital dynamics, "serosorting"
- Built-in epidemiology from EpiModel

### Friday

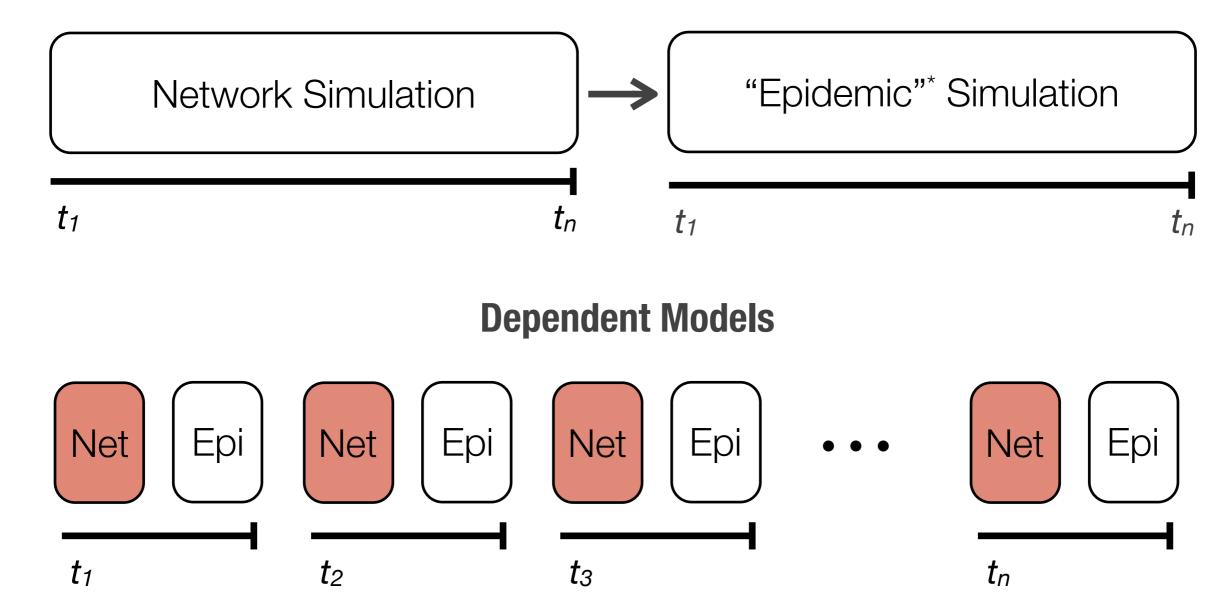
Extending EpiModel for your research questions

# Closed Population



### Model Dependence

#### **Independent Models**



<sup>&</sup>quot;Epidemic" = biological, behavioral, demographic, etc., changes

### EpiModel Workflow for Base Models

- 1. Construct the (empty) network data structure
- 2. Parameterize the TERGM (formation and dissolution formulas and target statistics)
- 3. Fit the TERGM, and diagnose the model fit
- 4. Parameterize the epidemic model
- 5. Simulate the epidemic
- 6. Analyze the simulation data

### EpiModel Workflow for Base Models

- 1. Construct the (empty) network data structure: network.initialize, set.vertex.attribute
- 2. Parameterize the TERGM (formation and dissolution formulas and target statistics): dissolution\_coefs
- 3. Fit the TERGM, and diagnose the model fit: netest, netdx
- 4. Parameterize the epidemic model: param.net, init.net, control.net
- 5. Simulate the epidemic: netsim
- 6. Analyze the model data: print, plot, summary, as.data.frame, ...

# Outline for Today

**Tutorial 1:** SIS Epidemic in "One-Mode" Network

Lecture: Considerations for Balancing



**Tutorial 2:** SIR Epidemic in "Bipartite" Network

**Exercise:** Translating Egocentric Data to Target Statistics





**Tutorial 3:** Time-Varying Biology & Behavior

**Tutorial 4:** Simple Interventions



Lab: Infectious Disease over Networks



# Lab Groups

- Today and tomorrow will feature long-form modeling labs in small groups
- You're in charge of forming a small group
- In picking your group, consider:
  - R abilities/experience
  - Prior network science and/or modeling experience
  - Population interests
  - Disease interests
- Ideal group size: 3 4

# Lab Groups

#### Cluster 1

HIV, HIV treatment, HIV in TGW, HIV in IDU, HIV + family planning, HIV PrEP

#### **Cluster 3**

Lemurs, elk, horses, monkeys (oh my!)

#### **Cluster 5**

Other human IDs: TB, HPV, HBV, enterics, influenza, malaria

#### Cluster 2

STDs, Syphilis, gonorrhea, STD resistance (NG), mycoplasma genitalium

#### **Cluster 4**

Plant diseases/systems

#### **Cluster 6**

Social science applications: communication and social diffusion, geography/migration