[In Progress]

EvoNetHIV User's Guide -- Software Package for Modeling HIV Epidemics and Viral Evolution in Sexual Networks

James Murphy, Steven Goodreau, Joshua Herbeck, and John Mittler

Version 1.0
November 15, 2017

University of Washington
Department of Microbiology
Department of Global Health
Center for Studies in Demography and Ecology

Preface

This document describes a R package (EvoNetHIV) that allows users to model the interactions between social networks, within-host dynamics, viral evolution, and HIV-1 spread. This package is intended for academic researchers in HIV epidemiology and evolutionary biology.

The package assumes familiarity with basic R commands. It is built using the API (application programming interface) of the R package *EpiModel*, so familiarity with that API and package is helpful. We provide an introduction below, as well as links to more information for advanced users. Users who wish to model new network structures will also need to be familiar with the methods and syntax from the *statnet* suite of R packages. We also provide a brief overview, and links for more information, within this Guide.

We acknowledge contributions from Neil Abernethy, Juandalyn Burke, Geoff Gottlieb, Kathryn Peebles, and And Sarah Stansfield.

This project was supported by a grant from the National Institutes of Allergy and Infectious Diseases Add grant number. (PIs Steven Goodreau, Joshua Herbeck and John Mittler) and the Center for Demography and Ecology. Findings from this research do not necessarily reflect the views of the funders.

For updates and further information about this package, please visit https://github.com/EvoNetHIV.

Table of Contents

[in progress]

1. EVONET OVERVIEW

Introduction

EvoNetHIV is a stochastic, agent-based simulation model that incorporates within-host dynamics, sexual network structure, viral load variation, treatment, condom use, circumcision, and HIV evolution. *EvoNetHIV* is written as a series of interchangeable modules and user-specified parameters (written in the R programming language) that control different components of the system (e.g. HIV and CD4 progression, treatment initiation, coital acts). Network characteristics (e.g., age-related homophily, presence of risk groups) can be specified using standard statnet network commands. Advanced users have the option of writing additional modules (e.g., allowing circumcision probabilities vary with risk membership).

Evonet foundations Evonet uses a modular framework from the EpiModel API (www.epimodel). The network module is built using exponential random graph models in the statnet framework (www.statnet.org), which provides a language for specifying arbitrarily complex network models.

Experimental framework. Users may interact with Evonet in one of four principal ways to conduct their own experiments: (1) they may keep the model structure exactly as is, but vary some of the many dozen input parameters; (2) they may use some of the alternative functional forms included with the model through the use of flag arguments; (2) they may keep the model structure as is, but vary the network model and associated parameters through the use of *ergm* terms; (3) they may change the structure of the model through the writing of substitute modules. Each of these features requires understanding the current model, and being familiar with the default parameter values; these are described below. Option (3) requires familiarity with the mathematics and terminology of exponential random graph models, and option (4) requires familiarity with the EpiModel API; an introduction to each of these can be found here: http://statnet.github.io/nme/

Agent-based framework. EvoNet is a stochastic, agent based simulation model. Each agent (i.e., individual) is a discrete entity that has over 30 attributes. Important agent attributes include sex, age, infection status, time of infection, viral load, set-point viral load, and treatment status. Some attributes are regularly updated, such as age or viral load (for infected agents) while others (e.g. sex, set-point viral load), do not change once assigned to the agent. Agents are "born" (added to the model as uninfected agents), age, and depart from the model, either through death from AIDS, death from background mortality, or aging out. Agents can become infected and/or infect other agents.

Stochastic framework. EvoNet is a stochastic model, such that identical parameter values or starting conditions for different model runs will produce different results. Many input parameters are mean or variance values of probability distributions used to draw random numbers from the specified probability distribution. For example, the parameter for mean sex acts per ongoing relationship per time step is the mean value for a draw from a Poisson distribution.

Daily timesteps. Model dynamics occur on a daily timestep; thus fine-scale epidemiological and behavioral processes can be simulated.

Epidemiological and evolutionary dynamics. EvoNet models have four main inter-connected components that result in epidemiological and evolutionary dynamics: *network structure*, *sexual*

behavior, clinical phenomena, and *SPVL and Viral progression.* All four components operate at the scale of the agent. The aggregate dynamics or values across agents give epidemiological and population level dynamics.

- ***Network structure*.** Agents can be either isolates or have one or more concurrent relationships with other agents. This network structure is dynamic and changes according to the specified network statistics.
- ***Behavior*.** Important behavioral dynamics include sex roles (for MSM), frequency of sex, condom use, HIV testing frequency, and adherence to treatment. Behavioral dynamics can be specified by the user.
    - ***MSM vs. Heterosexual models.*** MSM and heterosexual models are nearly identical. The most important differences are found in network structure. In MSM models, each agent is assigned either an "insertive", "receptive", or "versatile" role. Insertive agents can form relationships with versatile or receptive agents; receptive agents can form relationships with versatile or insertive agents; and versatile agents can form relationships with all three agent types. Heterosexual models assume a 50-50 sex ratio and only only relationships between opposite sexes are allowed.
- ***Clinical.*** *Treatment, CD4 stuff?*
- ***SPVL and Viral progression.*** Viral load progression is mostly a function of set-point viral load. SPVL for newly infected agents is determined by the donor's SPVL and by environmental (random) variation. The contribution of the donor's SPVL to the infectee's SPVL is determined by the assumed heritability of HIV SPVL.

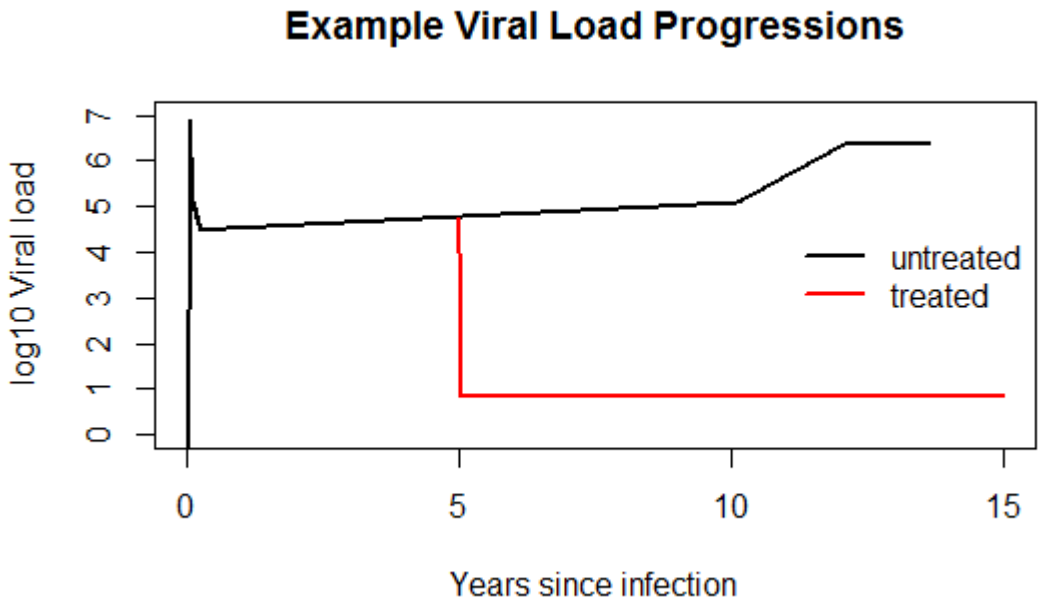


Figure 1. Example of viral load progression with and without ART. For this example, ART begins at 5 years since infection. Without ART, AIDS begins at about 10 years since infection and death at about 13 years. While acute phase duration is fixed at 90 days, start of AIDS and time of AIDS death are stochastic.

Virological, Epidemiological, and Demographic Processes

In this section we give an overview of the assumptions made by our default modules. Additional details and descriptions of alternative (nondefault) modules are given in the programming guide in section 2.

Viral Load and Setpoint Viral Load

- At infection, agents are assigned a set-point viral load (SPVL), using a function that incorporates infector SPVL, assumed heritability of SPVL, and a randomly determined host-specific factor (stochastic white noise), and heritability. The maximum possible viral load (all viral dynamics on log10 scale) is 7.0 and the minimum is 2.0.
- Initial VL starts out at -4.0 rises exponentially to a peak acute phase level, and then has a biphasic decay to SPVL. The acute phase last 50 days. Default value for peak acute phase VL is constant for all agents but can be set to be a function of an agent's SPVL. After the end of the acute phase, VL increases linearly, starting at SPVL, at a pre-specified chronic phase rate of increase. At the onset of AIDS, VL increases linearly at a pre-specified rate of increase for the AIDS phase until it reaches the assumed maximum VL value for AIDS at which it remains until agent's death. (The onset of AIDS is triggered by CD4 values described below.)
- With commencement of ART, VL will decrease linearly over 30 days to a "undetectable" level of 1.7 and remain there for the duration of treatment. If treatment stops, VL will then increase linearly over 30 days to the VL level at the start of treatment and then progress as if treatment did not occur.

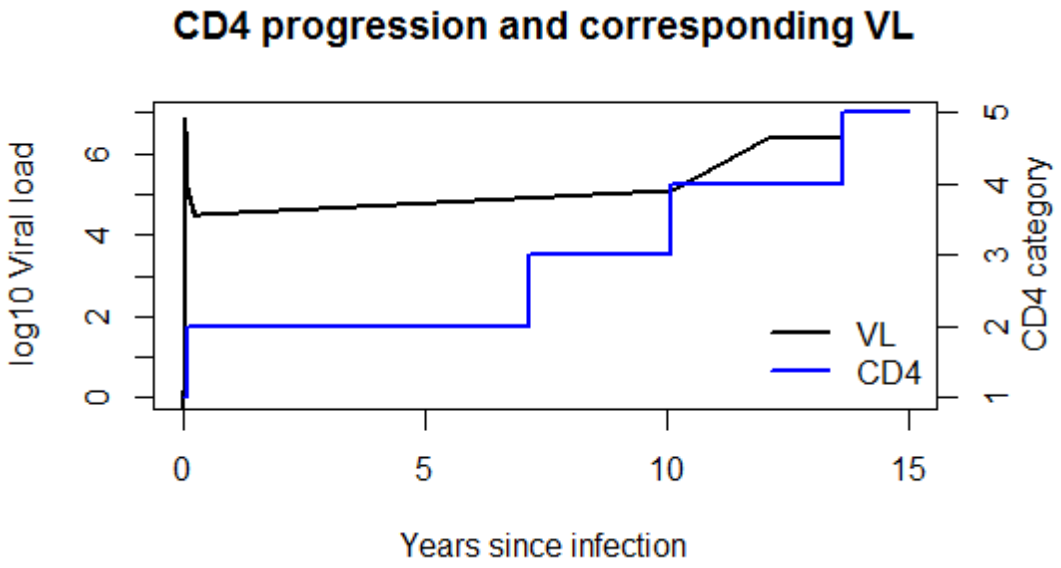


Figure 2. CD4 progression and corresponding V (no treatment).

CD4 Dynamics

- CD4 counts for an agent is a recorded as an ordinal variable with 4 categories; categories 1,2,3, and 4 represent CD4>500, 500>CD4>350, 350>CD4>200, 200>CD4>0, respectively. Category 4 represents AIDS stage.

- At infection, an agent is probabilistically assigned an initial CD4 category (1,2, or 3) where higher SPVL values increases the probability of assignment to categories 2 or 3. (Table 1). For each daily timestep, an infected agent in categories 1,2, or 3 can move to the following CD4 category (representing lower CD4 counts) or remain in the current category based on the outcome of a Bernoulli trial with the probability of success (moving to the next category) equal to the inverse of the mean passage time (in days) for a given SPVL and CD4 category (Table 2). Passage time in CD4 category 4 is fixed for each SPVL category; at the end of passage time in category 4, an agent is categorized as having died of AIDS.
- With commencement of ART, an agent's CD4 category is set to 1 and remains there until treatment stops or agent's death. If treatment stops before agent's death, the CD4 category of an agent is set equal to the value at the start of treatment.

s

Table 1. Probabilities of starting CD4 progression in categories 1,2, or 3.

Set point viral load (log_{10} copies/mL)	CD4 level (cells/mm^3) (category)		
	≥ 500 (1)	350 – 500 (2)	200 – 350 (3)
[2.0, 3.0]	0.88	0.12	0.00
(3.0, 3.5]	0.87	0.12	0.01
(3.5, 4.0]	0.85	0.12	0.03
(4.0, 4.5]	0.78	0.19	0.03
(4.5, 5.0]	0.73	0.21	0.05
(5.0, 5.5]	0.71	0.25	0.04
(5.5, 6.0]	0.64	0.27	0.09
(6.0, 6.5]	0.00	0.00	1.00
(6.5, 7.0]	0.00	0.00	1.00

Table 2. Mean passage times for categories 1,2, and 3. Fixed passage times for category 4.

Set point viral load (log_{10} copies/mL)	CD4 level (cells/mm^3) (category)			
	≥ 500 (1)	350 – 500 (2)	200 – 350 (3)	< 200 (4)
[2.0, 3.0]	6.08	5.01	3.60	4.67
(3.0, 3.5]	4.69	2.52	3.68	4.11
(3.5, 4.0]	3.94	4.07	2.38	3.54
(4.0, 4.5]	2.96	3.09	3.81	2.98
(4.5, 5.0]	2.25	2.32	3.21	2.42
(5.0, 5.5]	1.47	1.55	2.27	1.86
(5.5, 6.0]	0.95	1.19	1.00	1.29
(6.0, 6.5]	0.32	0.59	0.68	0.73
(6.5, 7.0]	0.30	0.46	0.37	0.17

SPVL, VL, and CD4 of initial population

- SPVL of ithe initial infected population is assumed to be normal distributed with a mean of 4.5 and variance of 0.8. SPVL values for initial population are drawn probabilistically from this distribution (user can change these values).

- Each model run begins with a specified number of infected agents. Each of these initially infected agents are randomly assigned an infection time 180-730 days before the start of the model run. Then VL and CD4 are projected forward based on infection time. Thus, each infected agent in the initial population will be in the chronic phase of HIV infection at the start of the model.

Table 3. VL and CD4 progression parameters

Description	Value	Unit
viral load at Day 1 of infection	0.0001	copies/ml
Viral Load at peak of Primary Infection	7.7 x 10^6	copies/ml
VL Maximum In AIDS	2400000	copies/ml
Daily Slope in VL Increase After AIDS Onset	1.0041122	unitless per day rate
Time To Peak VL during Acute Infection	21	Days
Time In Acute Inf	90	Days
Time until beginning of Acute Phase decline part 2	55	Days
Slope on ln scale in Acute Decline Phase 2	-0.03	NA
Slope of viral load increase during chronic phse	0.14	per year

Table 4. VL progression with treatment parameters.

Description	Value	Unit
Final VL After Complete Treatment	13	copies/ml
VL Undetectable Level	50	copies/ml
VL Exponential Decline Rate With Treatment	-0.6	per day

Table 5. SPVL and heritability parameters

Description	Value	Unit
Average Log SPVL for initial population	4.5	log10 copies/ml
Variance Log SPVL for initial population	0.8	NA
Mutation Variance	0.01	(log10 copies/ml)^2 / per year
donor-recipient heritability (h^2)	0.36	NA

Condom Use and Circumcision

- ***Condom use*:** Each sex act has a specified probability of condom use. Default is 0.50 but can be easily modified by user. An optional parameterization models condom use as a function of age and is described in the Treatment as Prevention Case Study in Section 2.xx.
- ***Circumcision:*** Each male agent has specified probability of having undergone circumcision. Default is 0.85 but can be easily modified by user.

Testing, Diagnosis, Treatment

- ***Testing and Diagnosis:*** Default testing interval is 1 year for both sexes, and sex specific intervals can be specified. All HIV+ agents that are tested receive a positive diagnosis, which is required for undergoing treatment.

- ***Treatment:*** Treatment campaigns begin at user specified start times (e.g., 5 years after start of model run). Once treatment campaign begins, all diagnosed agents are put on treatment, though this proportion can be modified. Agents in acute infections are assumed not to undergo treatment but this constraint can also be removed. Further constraints can be placed on treatment, such as requiring a minimum viral load or time since infection.

Coital Acts and Transmission

Sexual acts are determined among agents in a serodiscordant relationship at each daily time step. Among these partnerships, the number of sexual acts per partnership per daily time step is assigned according to a Poisson draw (mean = 0.43). Probability of sex can also be a declining function of age. Default condom use is determined for each sexual act with probability of 50% and default circumcision status is assigned to agents at model entry with 85% probability. When the infected partner is 47% past their expected time in AIDS, no sex occurs.

- For MSM models, agents are assigned one of three sexual roles: versatile, insertive, and receptive. Insertive agents only have sex with receptive agents and vice-versa. Versatile agents can have sex with any other agents. Intra-event versatility (when men switch roles and each engage in insertive and receptive anal intercourse (AI) during the same sexual contact) occurs with 40% probability when both partners are role versatile.

Transmission

HIV transmission probabilities are calculated as a function of relevant risk factors for each sex act according to characteristics of the sexual act and characteristics of the agents engaged in the sexual act. Following Hughes et al., the probability of transmission is calculated for each sexual act that occurs in a serodiscordant relationship, as:

$$P(transmission) = 1 - (1 - \lambda)^{e^{X\beta}}$$

where λ is the baseline per-act infectivity (0.000247).

Heterosexual transmission
For heterosexual partners,

$$X\beta = \ln(2.89) * (viral\ load - 4.0) + \ln(RR_{circumcised}) * circumcised + \ln(RR_{condom}) * condom + \ln(RR_{age}) * age_{base} + \ln(RR_{sti}) * sti\_status\_sus$$

where *circumcised* is an indicator variable for the circumcision status of the male, *condom* is an indicator variable for condom use for the male, *viral load* is the log10 viral load of infected agent, and *age_base* is the number of decades over 35 the susceptible agent is: (age-35)/10; and RR_x is the risk factor.

MSM Transmission
For MSM partners,

$$X\beta = \ln(RR_{VL}) * (viral\ load - 4.0) + \ln(RR_{insertive}) * insertive + \ln(RR_{receptive}) * receptive + \ln((RR_{circumcised})) * circumcised + \ln(RR_{condom}) * condom +$$

$$\ln(RR_{sti}) * sti\_status\_sus$$

where *insertive* and *receptive* are indicator variables for role of susceptible agent and other variables are as for heterosexual partners.

Once the probability of transmission is calculated, a random uniform draw determines whether infection occurred.

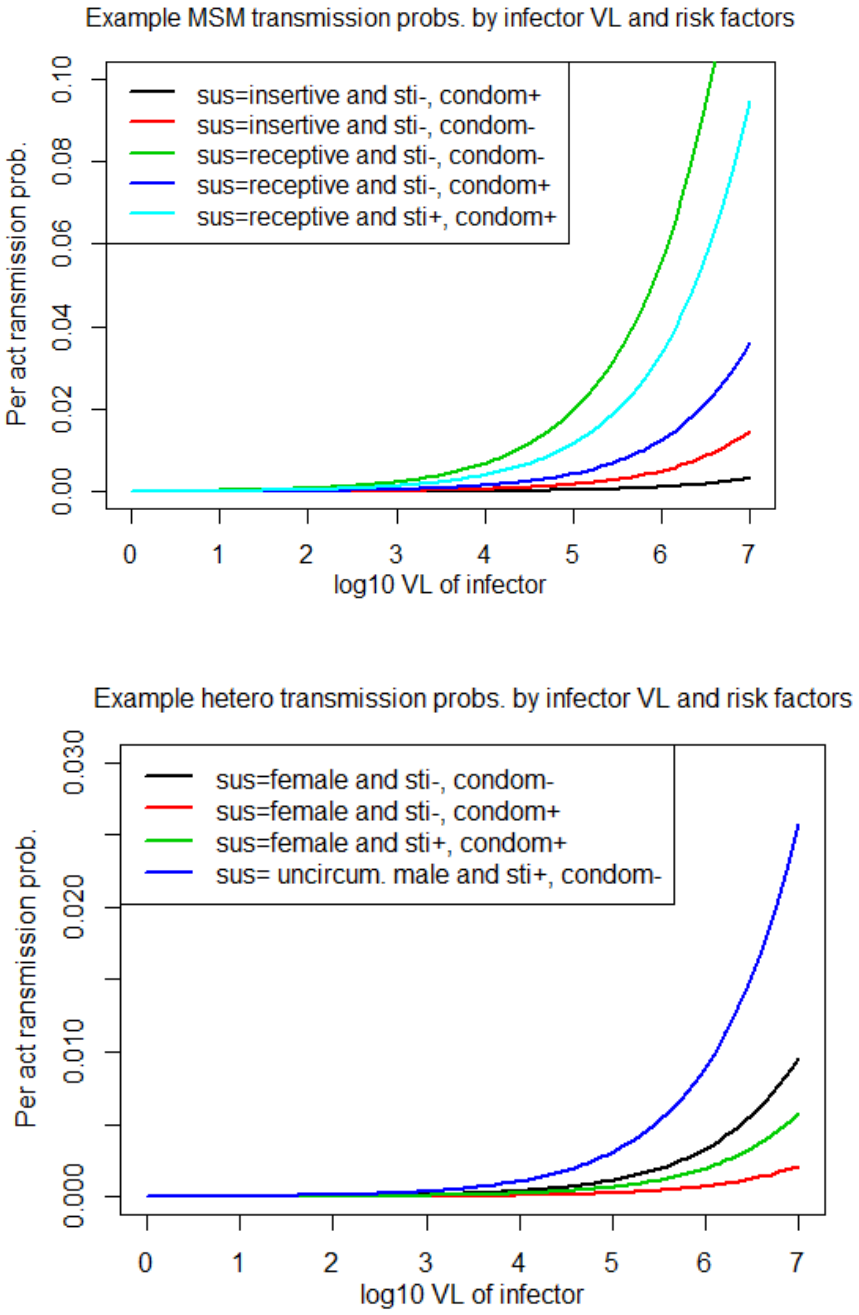


Figure 3. Example of transmission probabities as function of risk factors for MSM (top) and heterosexual partners (bottom).

Set Point Viral Load

Set point viral load (SPVL) in infected agents at model initialization is generated as a combination of viral (viral genotype) and environmental (a combination of undefined host and non-viral) factors. For infected agent *i* present at the start of the model, the viral contribution to SPVL is drawn from a normal distribution:

$$viral_{spvl,i} \sim N\left(\mu_{spvl,t_0}, \sigma^2_{viral,t_0}\right)$$
$$\mu_{spvl,t_0} = 4.5 \log_{10} \text{copies/mL}$$
$$\sigma^2_{viral,t_0} = h^2\, \sigma^2_{spvl,t_0},$$

where h^2 is the heritability coefficient, set here at 0.36, following Hollingsworth et al. (2010), and consistent with the Fraser et al. meta-analysis (2014), and σ^2_{spvl,t_0} is the variance of the distribution of SPVL in the initial population at model start. Note that this while the value of h^2 is set as a model input and employed to modulate the influence of viral genotype on the similarity in SPVL between transmission pairs, heritability is a population-level measure that can change over time and across populations.

For infections after model start, the viral component for newly infected agent *i* is:

$$viral_{spvl,i} = viral_{spvl,infector} + \epsilon$$

where ϵ is the assumed normally distributed stochastic mutational variance, $\epsilon \sim N(0,\ 1e-4)$.

The stochastic environmental contribution is normally distributed and is calculated similarly for the initial population and subsequent infections:

$$env_{spvl,i} \sim N(0, \sigma^2_{env})$$
$$\sigma^2_{env} = (1 - h^2)\sigma^2_{spvl,t_0}.$$

SPVL is then the sum of the viral and environmental contributions, constrained to a minimum value of 2 $\log_{10}$ copies/mL and a maximum value of 7 $\log_{10}$ copies/mL:

$$spvl_i = env_{spvl,i} + viral_{spvl,i}.$$

Demographics

- **Aging:** age of each agent is updated each daily timestep. Default minimum age is 18 and default maximum age is 55.
- **Aging out:** agent is removed from population when maximum age is reached.
- **New agents ("births"):** New agents are added to the population assuming a 1% growth rate (in absence of an epidemic); for heterosexual model, new agents have a 50/50 sex ratio. Default age of new agents is 18.
- **Deaths:** Infected agents can die of
    - AIDS – when their passage time through CD4 category 4 is completed;
    - Natural mortality, independent of HIV – default sex and age specific natural mortality values are from recent U.S. census data
    - Natural mortality, HIV induced – infected agents have a heightened risk of non-AIDS death

- **Initial age distribution:** Default sex-specific initial age distribution is based on U.S. census data

Network Structure and Dynamics

- **Structure:** User specified network structure based on the statnet and EpiModel packages can be implemented (e.g., mean degree, age and other subgroup structure). Default structure is random mixing of agents with a network mean degree of 0.7.
- **Partnerships:** Each agent can exist as an isolate (no-partners) or have one or more sexual partners per timestep. Default mean partnership duration is 100 days.
- **Risk groups:** Various risk group substructures can be imposed.

Current Research

[in progress]

Future Research Topics

[in progress]

Drug resistance evolution

We are developing a new within-host module that uses stochastic differential equations to model the concentrations of antirerotroviral drugs and and wildtype and drug resistance viruses within each agent. The new module updates concentrations of drugs and viruses multiple times per day. At the end of each day, updated values are stored as agent-specific attributes. We are currently using this new routine to explore the effects of therapy interruptions on the evolution of drug resistance. In future studies, we will use this code to explore the effects of risk-groups on drug resistance. We hypothesize that drug resistance viruses will spread faster in populations in which people with poor adherence patterns preferentially associate with each other.

2. USING EVONET

Installing and running EvoNet (with simple example)

Download/install/load EvoNet from GitHub website.

```
if (!require("devtools")) install.packages("devtools")
install_github("EvoNetHIV/EvoNet",subdir="pkg")
library(evonet)
```

Load default parameters.

```
evoparams <- evonet_setup()
```

Change default parameters. In this example, we change initial population size to 200 (from 100); number of initially infected agents to 40 (from 20); and the duration of the model to 20 years.

```
evoparams$initial_pop      = 200
evoparams$initial_infected = 40
evoparams$n_steps = 365*20
```

Create initial network (as a function of input parameters)

```
nw <- nw_setup(evoparams)
```

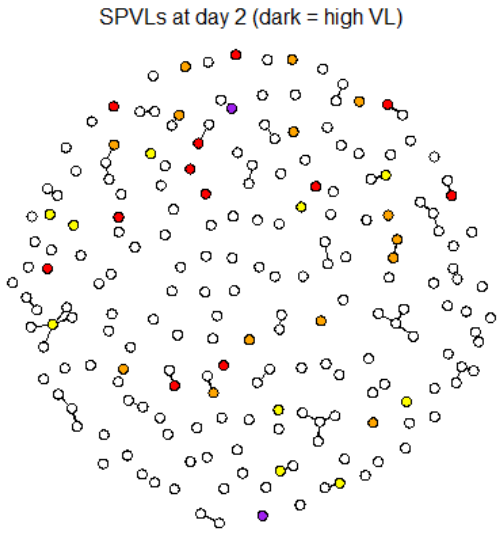
Specify modules/functions (as character strings) to simulate epidemic with desired dynamics.

```
modules <- c(
"aging",
"testing",
"treatment",
"viral_update",
"coital_acts",
"transmission",
"deaths",
"births",
"summary")
```

Run the simulation.

```
evomodel <- evorun(modules,evoparams,nw)
```

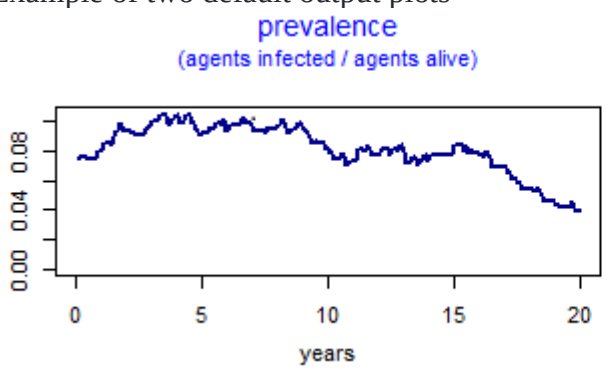
Initial network plot will be plotted by default at simulation start (colored circles represent infected agents). (Day 2 is first day of simulation as model initialization is considered day 1.)

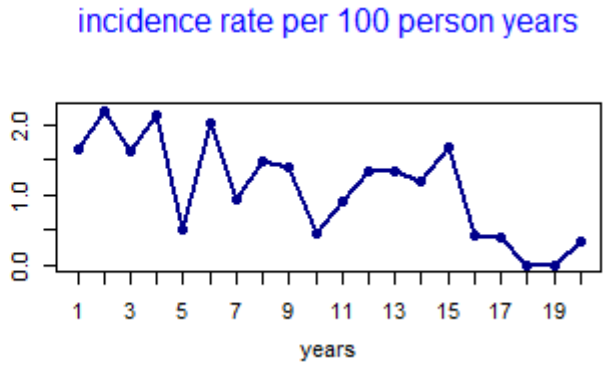


Plot summary figures (automatically to screen and saved as pdf)

```
evoplot(model=evomodel)
```

Example of two default output plots





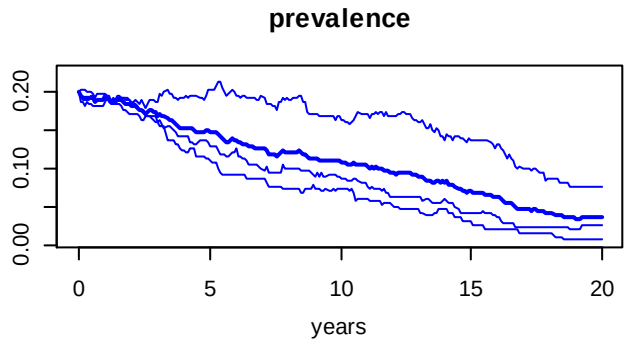
Approximately 30 default output plots are produced at the end of each model run. These plots are both printed to the screen and saved as a pdf file. The name of the pdf file and its location path can be specided with

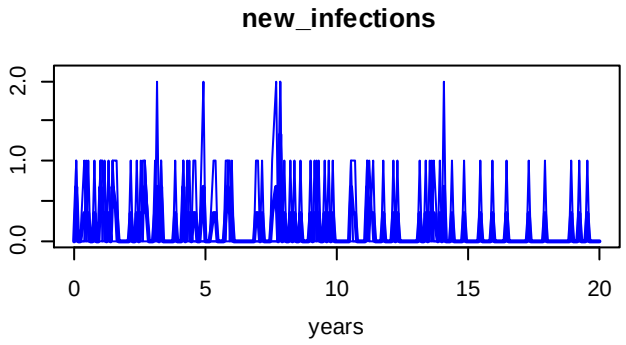
```
evoplot(model=evomodel, names = "…", path= "….")
```

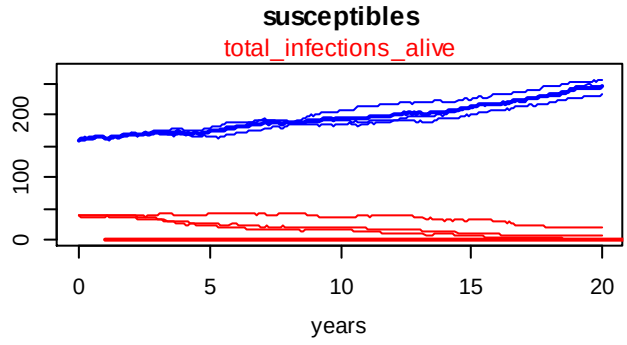
By default the pdf is called "evoplots.pldf" and is placed in the current working directory. The next several pages show the default output plots.

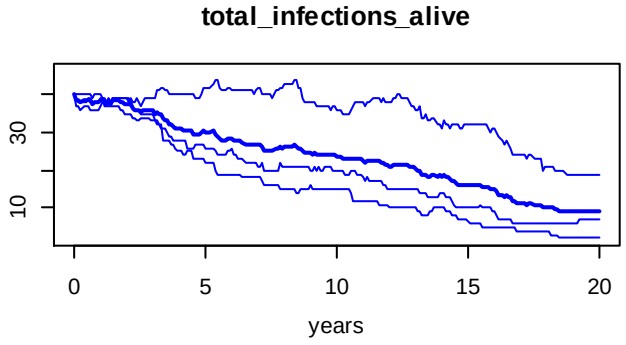
The following eight pages show the default output plots. The contents of each plot is further explained in Section x.x. The model for these plots had 3 simulations (or replicates) by setting the parameter "nsims" to 3: e.,g "evoparams$nsims=3 "– the default value is 1.

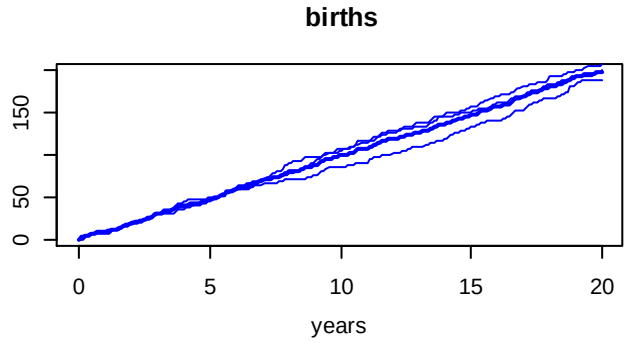
Default output plots for MSM model.



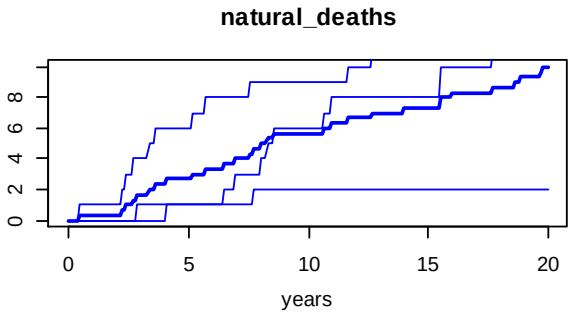


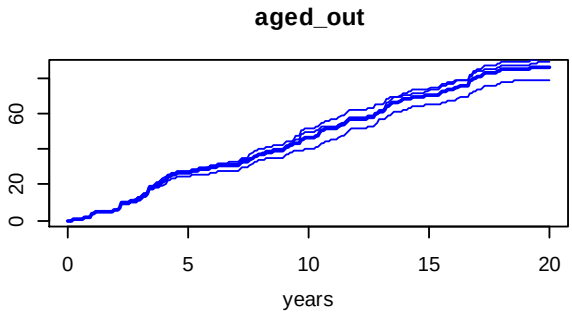


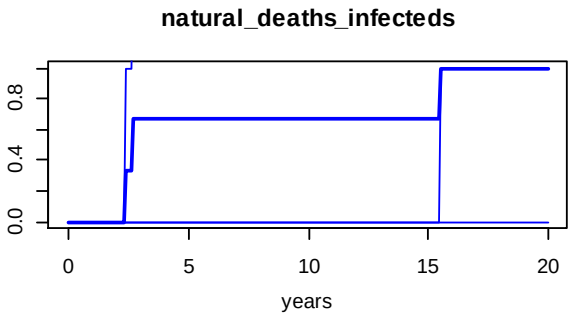


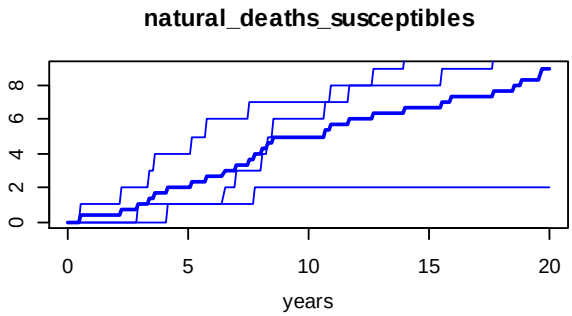


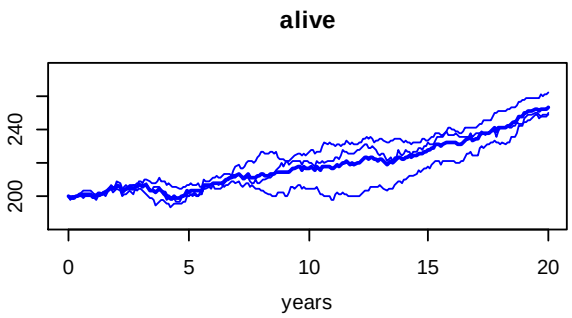
Default output plots for MSM model cont.

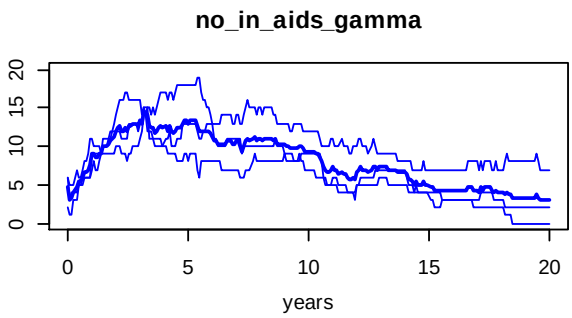




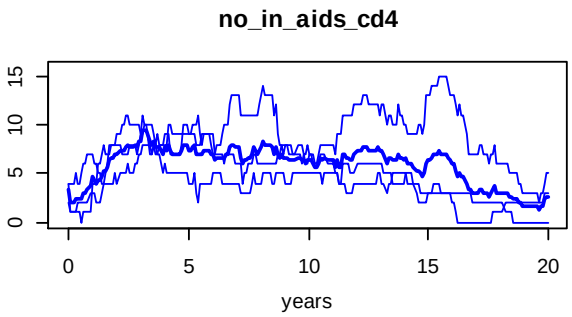


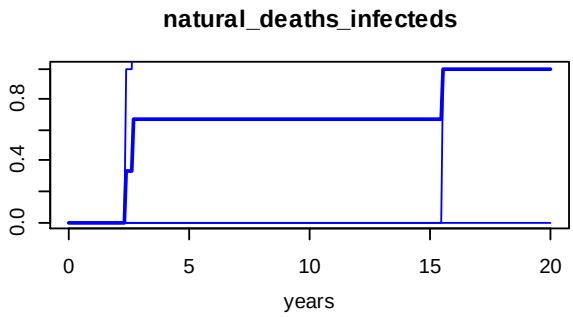


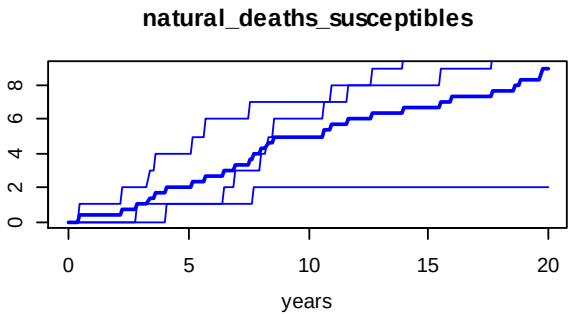


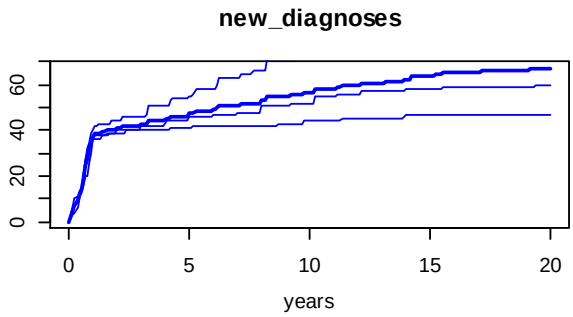


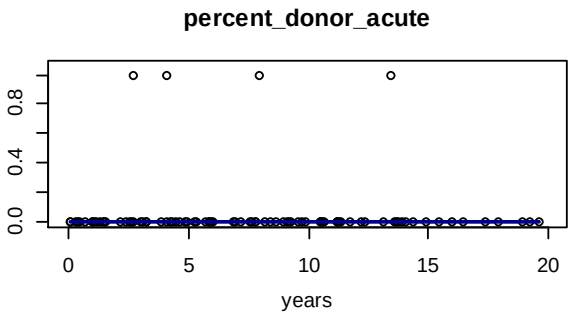
Default output plots for MSM model cont.

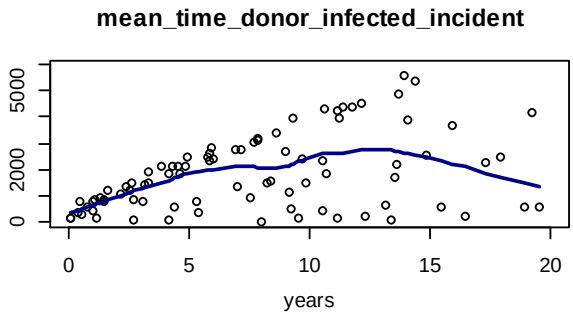




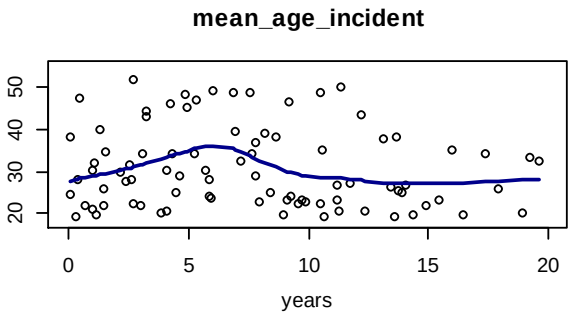


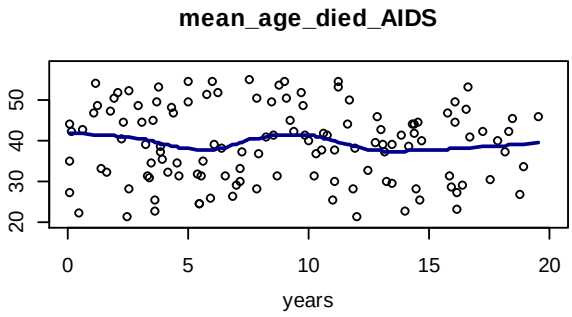


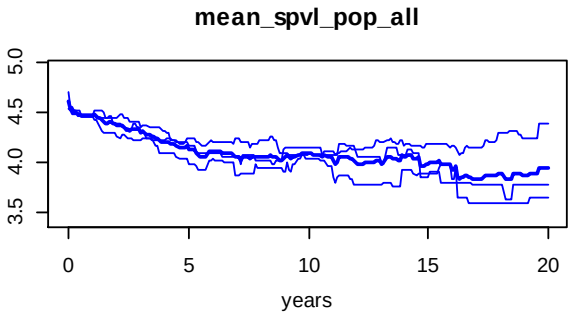


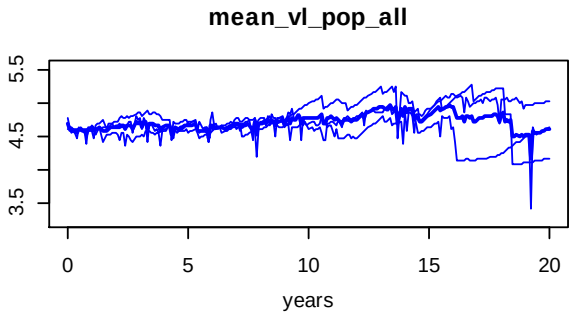


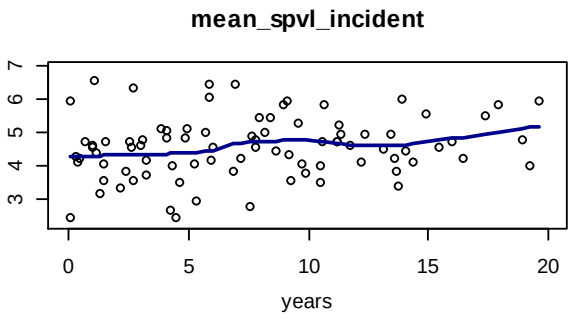
Default output plots for MSM model cont.

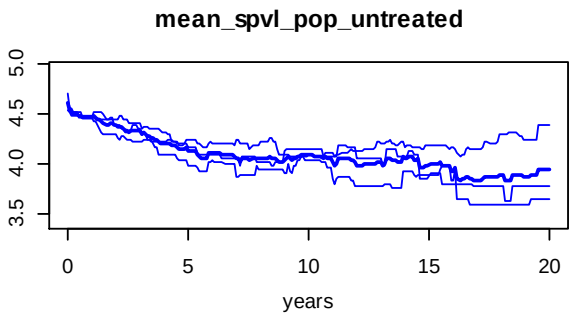




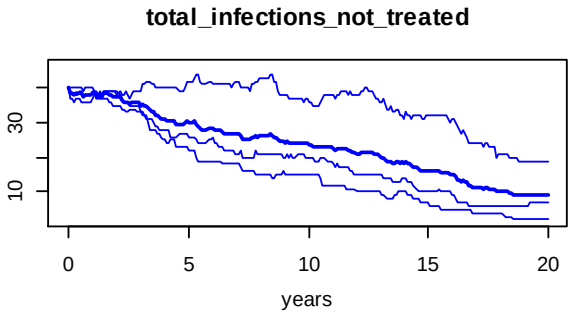


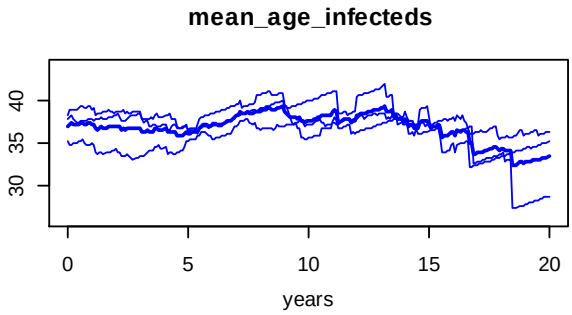


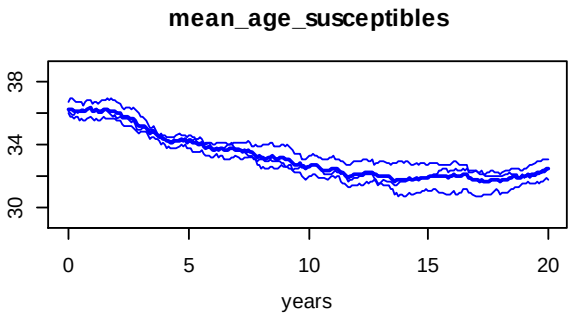


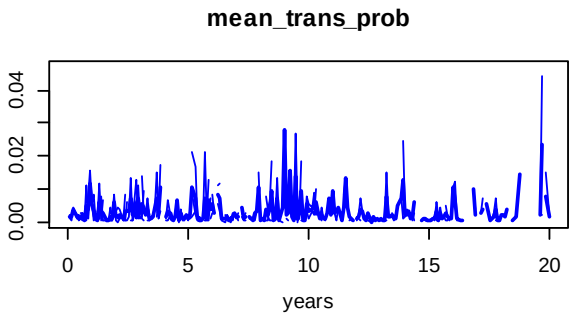


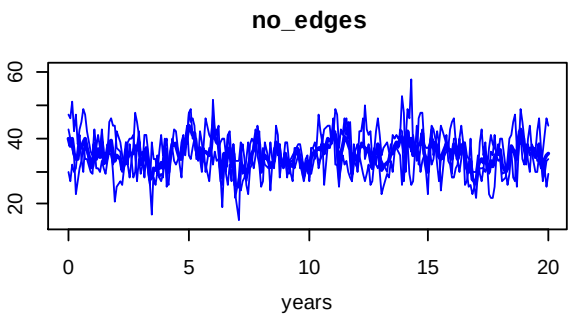
Default output plots for MSM model cont.

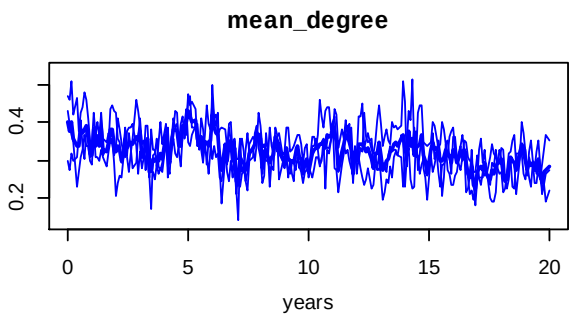




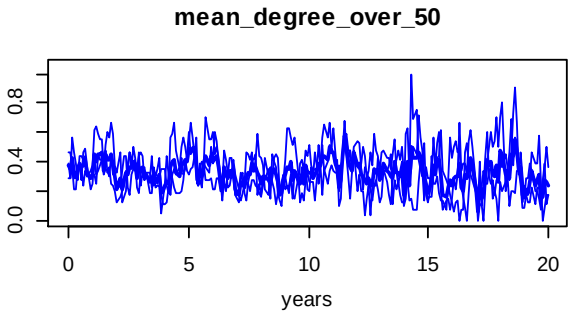


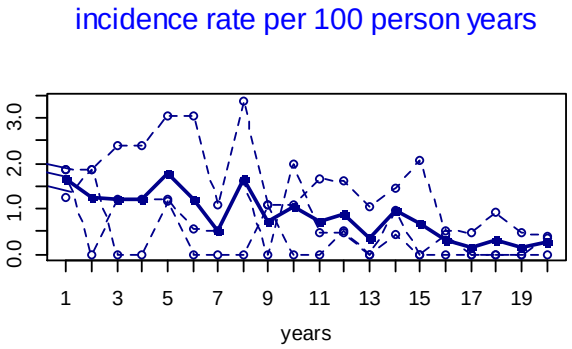


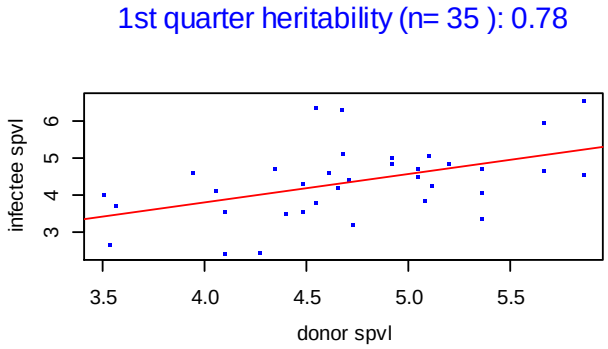


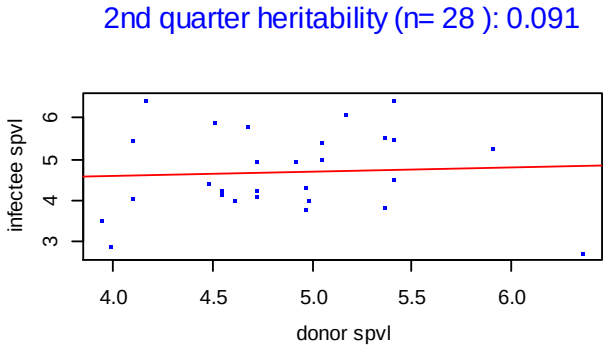


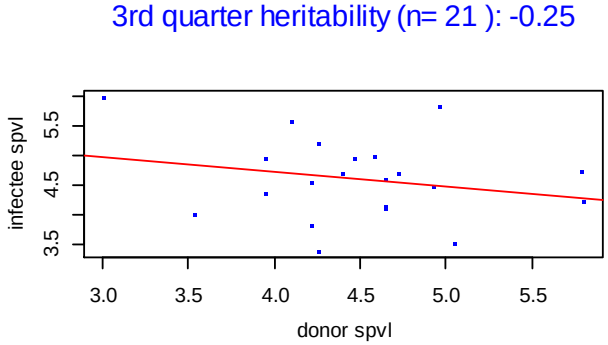
Default output plots for MSM model cont.

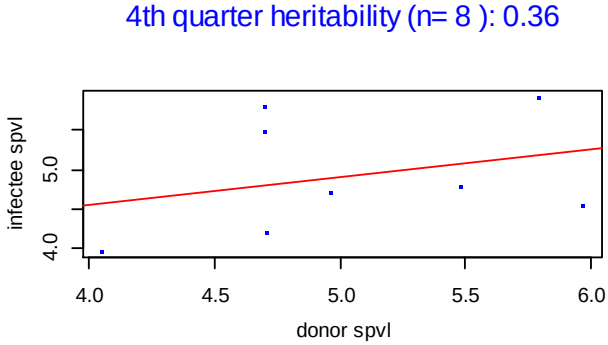




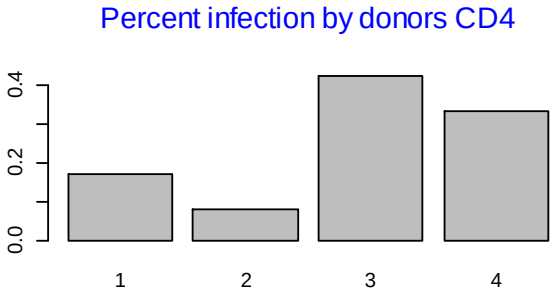


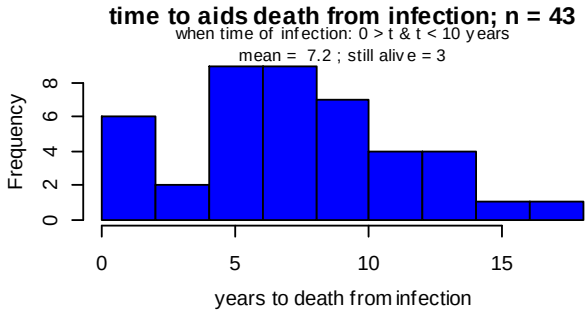


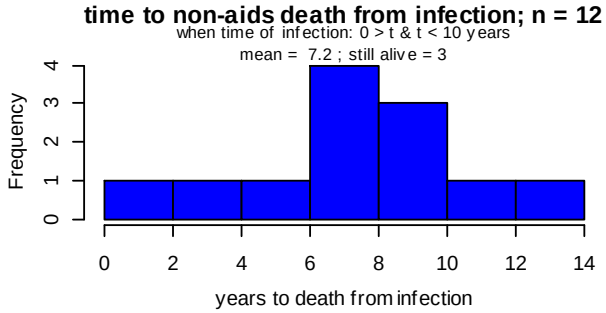


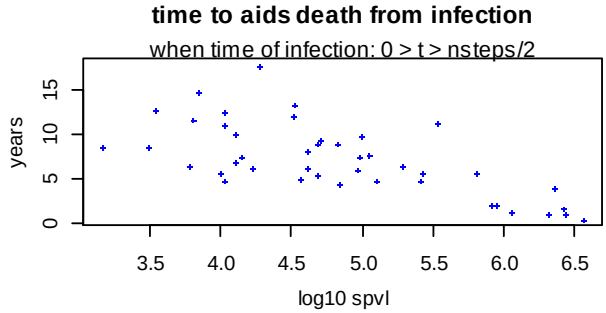


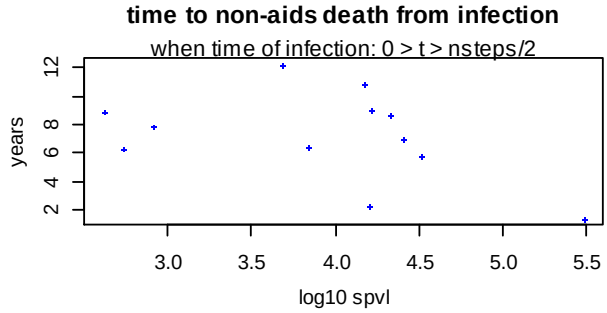
Default output plots for MSM model cont.

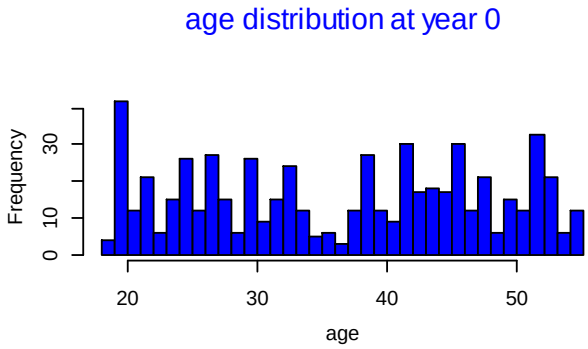




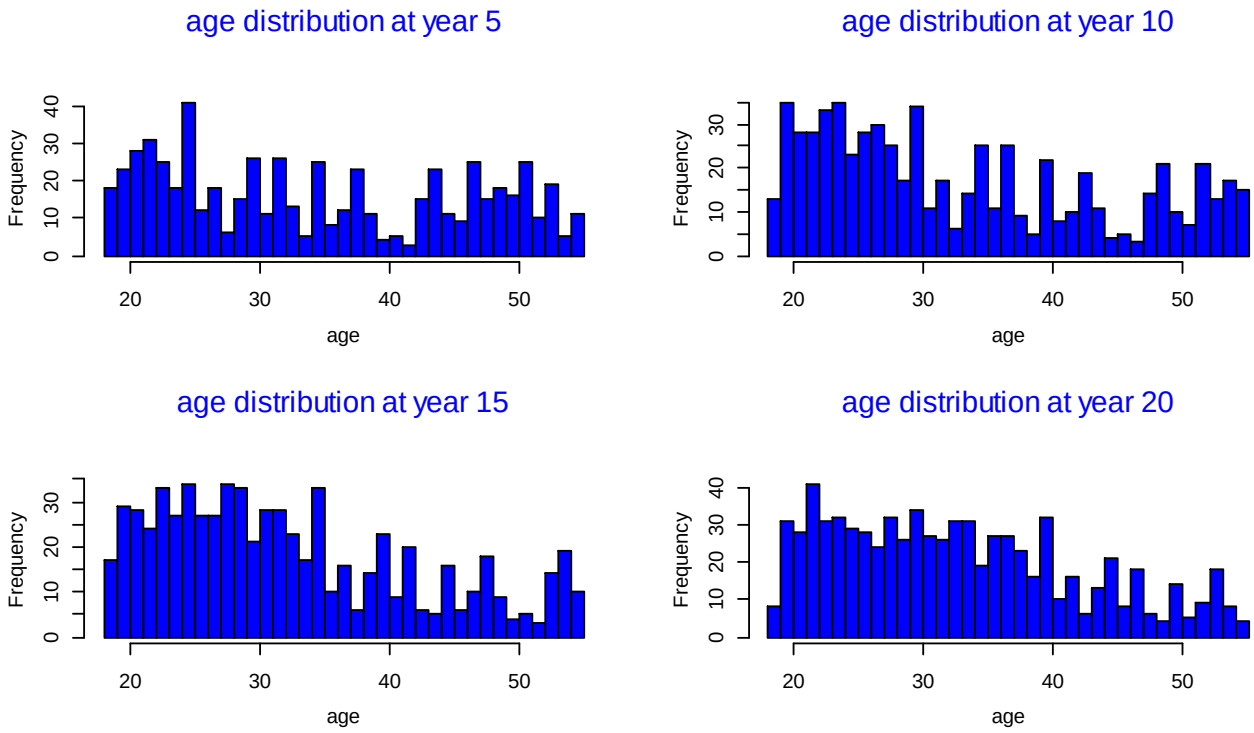








Default output plots for MSM model cont.



Basic model structure

Agent attributes

Each agent has an associated set of variables that describe/quantify demographic, behavioral, and epidemiological conditions or states. The most important attributes are sex, age, sex role (for MSM), circumcision status, infection status, treatment status, SPVL, and VL. Some attributes remain fixed for the duration of the model run (e.g., sex) while others are updated each timestep (e.g., age and VL). A full list of agent attributes and their default values are given in Section xxx.

Sequence of Virological, Epidemiological, and Demographic Processes

Model dynamics are specificed by a set of "modules" which execute a specific process: e.g., the "aging" module updates each agent's age per timestep. Default modules are briefly described below and in more detail in later sections. Users can write their own modules (either as modifications of existing ones or completely new).

- **aging:** updates each agent's age one day
- **testing:** determines whether an agent will be tested that timestep; tested and infected agents are then assumed diagnosed as HIV+.
- **treatment:** if a treatment campaign has started and diagnosed agents meet the specified criteria for receiving treatment, an agent will start treatment; the primary consequence of being on treatment is a decline of VL to a pre-specified undetectable level.

- **vl_update:** updates the VL of infected agents each timestep until death or initiation of treatment
- **update_cd4:** updates the the CD4 category of infected agents each timestep until death or initiation of treatment
- **coital_acts:** for each current relationship, determines the number of sexual acts per timestep, condom use, MSM sexual role,
- **transmission:** for each serodiscordonant relationship with a sexual act in a given timestep, calculates the probability of infection based on risk factors of partners; for newly infected agents, associated SPVL, VL, and CD4 values are calculated.
- **deaths:** labels agents as dead either due to AIDS and natural morality; also, removes agents that have aged-out of specified model age range (default 18-55 years old).
- **births:** adds new agents to the model based on assumed 1% annual growth rate of initial population
- **summary:** calculates population level summary statistics based on individual agent attrbitues (e.g., prevalence).

MSM vs. Heterosexual models

MSM and heterosexual models are nearly identical. The major difference is that MSM agents have a "role" attribute that deteremines whether they are receptive or insertive for a given sex act. Default model structure assumes an MSM model. To switch from default MSM network model to a heterosexual model, the following three parameters are changed to the following values: "model_sex", "nw_form_terms" (network formation terms), "nw_coef_form" (network coefficients for formation terms). This changes the default model from a MSM "edges only" (random mixing) to a heterosexual edges only model. Further specification of network structure is described below.

```
#read in default parameters (as list)
evoparams <- evonet_setup()
#override default values
evoparams$model_sex <- "hetero"                    #Defaault = "msm"
evoparams$nw_form_terms <- "~edges + offset(nodematch('sex',diff=F))"  #Default = "~edges"
evoparams$nw_coef_form <- -Inf                     #Note negative sign
```

Model Outputs

The most important model outputs are time-series of population-level summary statistics calculated each timestep from the values of all active (alive) agents. Of these, time series of prevalence, incidence, mean SPVL, and number of AIDS deaths will likely be the most informative. At the end of each model run, these time series are plotted to screen and to a pdf file. These time series are also part of the final model object and can be queried for specific values or for further analysis (see Section xxx). Additionally, the data (attributes) for each agent are saved and can be queried to answer very specific questions (e.g., how many women were 20-30 years of age when infected?), also disussed further in Section xxx.

Network structure

The default network structure is an MSM "edges only" (radom mixing). Specification of a heterosexual edges-only model was described above. EvoNet relies on the EpiModel interface to the statnet suite of ergm (exponential random graph model) related functions. Typically, changing the "nw_form_terms" and "nw_coef_form" parameter values (as in the above heterosexual example) will allow for various desired network structures to be implemented. The relationship duration parameter

“relation_dur” also determines the mean duration of each relationship – default value is 100 days but can be modified as needed.

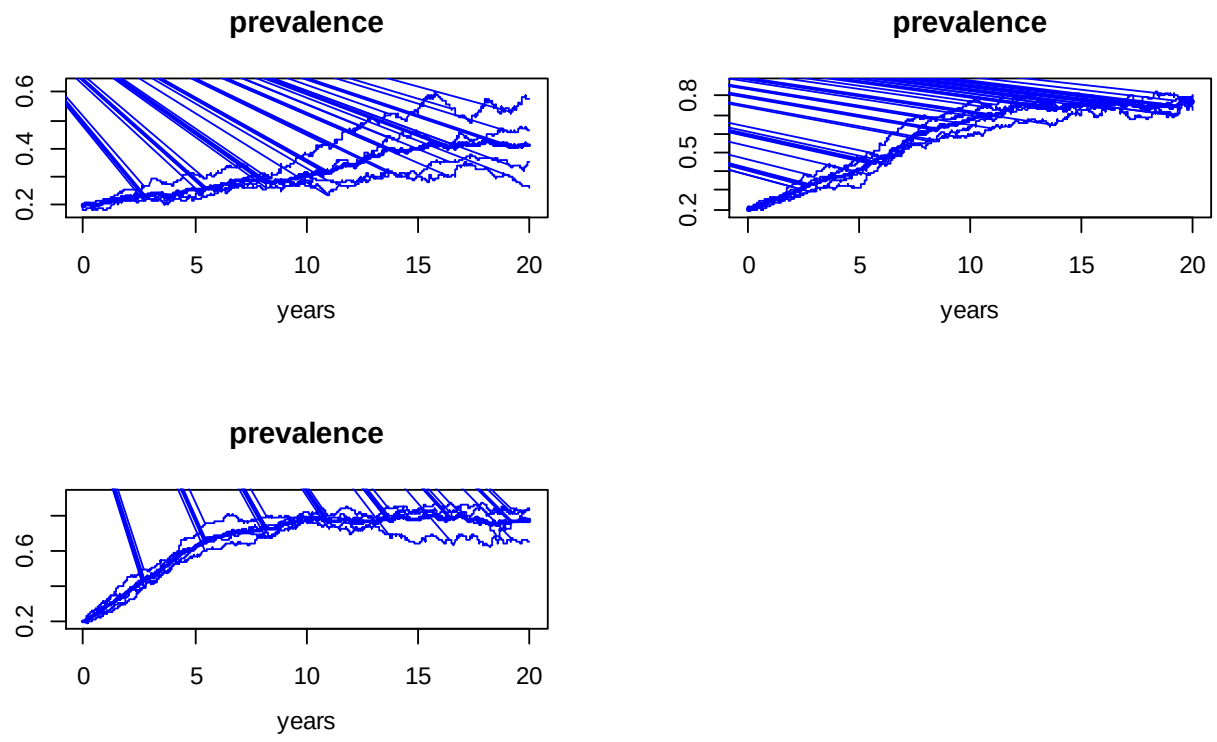
The EpiModel website for Network Models provides a basic introduction how to create various network structures: http://statnet.github.io/tut/BasicNet.html. As EvoNet uses the EpiModel modeling framework, EpiModel materials on network estimation is directly applicable to Evonet.

Using Loops

Likely, the effects of a range of parameter values on epidemic dynamics will be of interest. This will require using a loop structure. A simple example is given below where three values for the parameter “target_stats” is given. “target stats” explanation ….

```
#Read in default parameters (as list)
evoparams <- evonet_setup()

# Override default values
evoparams$initial_pop      = 200
evoparams$initial_infected = 40
evoparams$n_steps = 365*20
evoparams$nsims=4 # 4 simulations (replicates) for each target stat value

#Use default modules
modules <- c(
  "aging",
  "testing",
  "treatment",
  "viral_update",
  "coital_acts",
  "transmission",
  "deaths",
  "births",
  "summary_module")

#Three values of target stats, corresponding to mean degrees of 0.7, 1.0 and 2.0.
target_stats_values <- c(0.35,0.7,1.0)*evoparams$initial_pop
#Create list to hold output of three model runs
output_list <- vector('list',length=length(target_stats_values))

#Run model for each value of specified target stats
for(ii in 1:length(target_stats_values)){
  evoparams$target_stats <- target_stats_values[ii]
  nw <- nw_setup(evoparams)
  output_list[[ii]] <- evorun(modules,evoparams,nw)
}
```

```
# Plot and compare prevalence time series for each target stat value (prevalence increases with increasing target stat # / mean degree values)
par(mfrow=c(2,2))
evoplot(output_list[[1]],variables = "prevalence")
evoplot(output_list[[2]],variables = "prevalence")
evoplot(output_list[[3]],variables = "prevalence")
```




```
#Plot all output for first model
evoplot(output_list[[1]])
```

Reading in default parameters

Evonet parameters are read in as a list using the “evonet_setup” function. Parameters are a mixture of single numeric values, a vector or matrix of numeric values, or a character string. Changing default parameter values is crucial to running simulations. In this guide, evonet parameter values are stored in the list object “evoparams”; however, any name can be used. Alternatives one and two will catch any misspelling (typos) of parameter names (e.g., if parameter “initial_pop” is typed in as “initial_pup”, an error message wil be returned.)

```
#To view names of all evonet parameters
names(evonet_setup)
sort(names(evonet_setup)) #alphabetical order
```

```
#To view default parameter values
evoparams <- evonet_setup
evoparams$nsims #how many replicates
evoparams$nsteps # length of simulation in days
evoparams$intial_pop # size of initial population

#Reading in parameters: Alternative 1, using a list and the do.call function
param_list=list(
  initial_pop = 400,
  initial_infected = 40,
)

evoparams <- do.call(evonet_setup,param_list)

##Reading in parameters: Alternative 2: using arguments directly in the evonet_setup function
evoparams <- evonet_setup(initial_pop=400,
                          initial_infected=20)

##Reading in parameters: Alternative 3: readin all default parameters to "evoparams" object
#then modify/override
evoparams <- evonet_setup()
evoparams$nsim=1
evoparams$initial_pop=400
```

Simple example with concurrency

Blurb about concurrency and its implemenatation here.

```
#Read in default parameters
evoparams <- evonet_setup()

# Override default values
evoparams$nsims=3
evoparams$popsumm_frequency =30
evoparams$n_steps           = 365*30
evoparams$initial_pop       = 300
evoparams$initial_infected  = 30

#Use default modules
modules <- c(
  "aging",
  "testing",
  "treatment",
  "viral_update",
  "coital_acts",
  "transmission",
  "deaths",
  "births",
  "summary_module")
```

```
#create two variable vectors, each of length 2

#i) network formation terms: 1) random mixing with MSM role constraints 2) same as one but
# also with concurrency

nw_form_terms_vector=c( "~edges + offset(nodematch('role', diff=TRUE, keep=1:2))",
  "~edges + concurrent + offset(nodematch('role', diff=TRUE, keep=1:2))")

# ii) target stats: random mixing mode needs only one target stats value, concurrency model needs two
target_stats_list=list(0.35*evoparams$initial_pop,
                c(0.35,0.2)*evoparams$initial_pop)

#Create list to hold output of three model runs
output_list <- vector('list',length=length(nw_form_terms_vector))

#Run model for each value of specified target stats
for(ii in 1:length(nw_form_terms_vector)){
  evoparams$nw_form_terms  <-  nw_form_terms_vector[ii]
  evoparams$target_stats <- target_stats_list[[ii]]
  nw <- nw_setup(evoparams)
  output_list[[ii]] <- evorun(modules,evoparams,nw)
}

save(output_list,file="concurrSim.RData")

# Plot and compare prevalence time series for each target stat value (prevalence increases with increasing target stat
# / mean degree values)
par(mfrow=c(2,2))
evoplot(output_list[[1]],variables = "prevalence")
evoplot(output_list[[2]],variables = "prevalence")

#Plot all output for each model
evoplot(output_list[[1]])
evoplot(output_list[[2]])
```