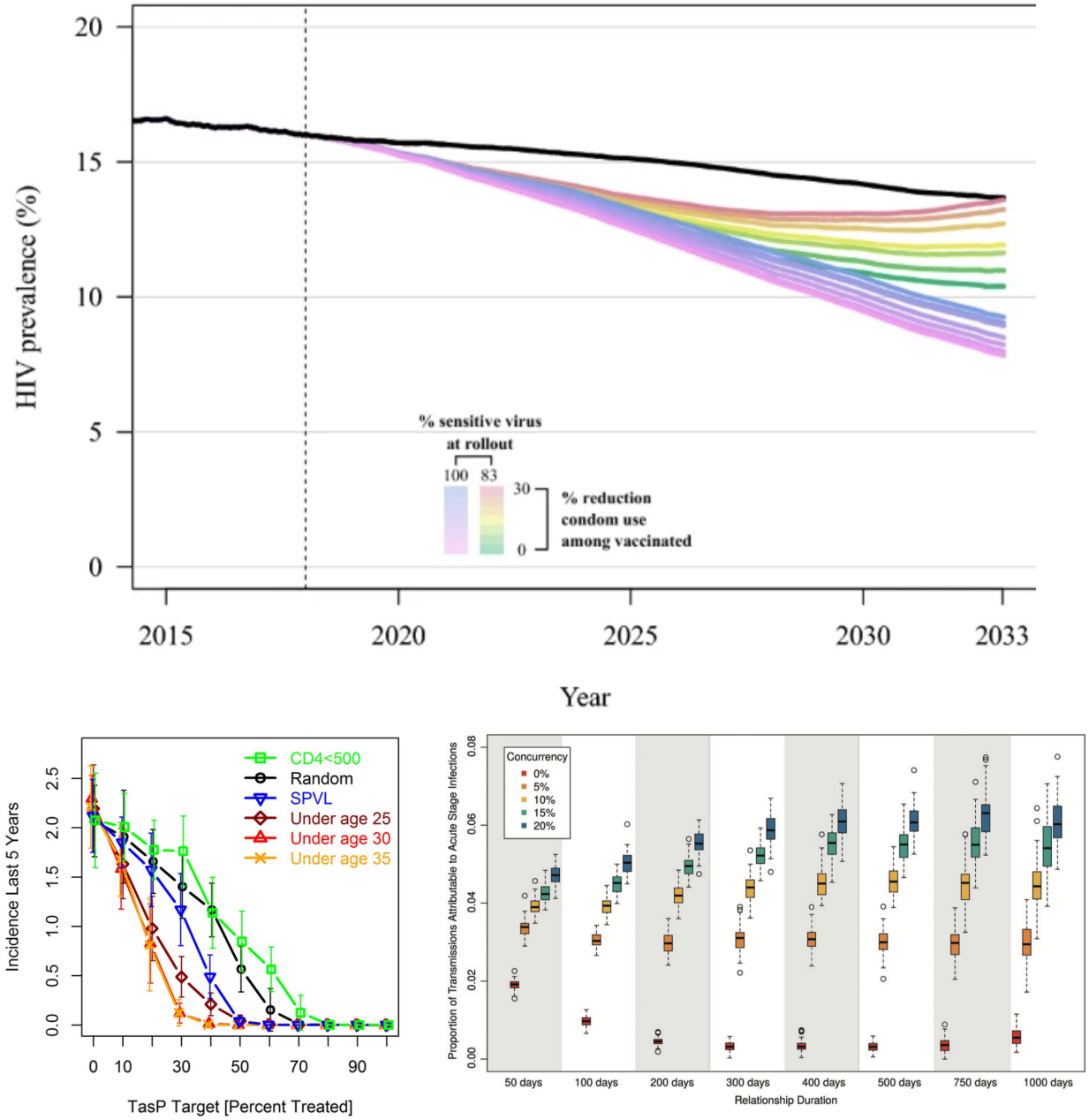
EvoNetHIV Quick-Start Users Guide

7/9/2021

This guide is intended for users who want to learn how to quickly install and run simple evonet programs.

I. Overview

EvoNetHIV is a stochastic agent-based model that simulates the spread and evolution of HIV-1 in sexual networks in which partnerships form and dissolve over time. EvoNetHIV is written in the R programming language as a series of modules that specify things like treatment, vaccination, network structure, sexual behavior, and within-host dynamics. Each agent (i.e., individual) is a discrete entity that has attributes such as sex, age, HIV infection status, time of infection, viral load, and treatment status. The software has ~400 user-modifiable parameters and saves timecourse data and data each agent's attributes at the end of the simulation.



Example EvoNetHIV outputs: Top: Prevalence during vaccination campaigns (Peebles et al. 2021). Bottom left: Final incidence after different TasP campaigns (Mittler et al. 2019). Bottom right: Percent infections during the acute phase under different assumptions about concurrency and relationship duration (Goodreau et al. 2018).

II. Installing and running EvoNetHIV

Step 1: Download and install EvoNetHIV. Easiest to do with a newly opened instance of RStudio or R. If installing for the first time, you may have to wait a few minutes for install_github to install lower-level packages that EvoNetHIV depends on.

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

Step 2: Load default parameters. This command will create a R list giving the default values of all the EvoNetHIV parameters.

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

Step 3: 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 run time 5 years.

```
evoparams$initial_pop      <- 200
evoparams$initial_infected <- 40
evoparams$n_steps          <- 365*5
evoparams$nsims            <- 2      # Perform two replicates
```

Step 4: Create the initial network. This network will be a function of the parameters given in the parameter list evoparams.

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

Step 5: Specify the EvoNetHIV modules (functions) you want to use. In the course of working with EvoNetHIV, the developers created roughly 40 different modules to account for different scenarios. In this example, we specify the base modules that we have used for most of our simulations.

```
modules <- c(
    "aging",
    "testing",
    "treatment",
    "viral_update",
    "coital_acts",
    "transmission",
    "evo_departures",
    "evo_arrivals",
    "summary_module")
```

Step 6: Run the simulation (save output to a name of your choosing. In this case "evomodel").

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

Step 7: Save model output. Default location is current working directory (use getwd() to identify).

```
    save(evomodel, file ="evomodel.RData"))
```

Step 8: Plot time course data To see the number of infected and susceptible people, type

```
plot(evomodel, y = "total_infections_alive", sim.lines = TRUE,ylab="Infected")
plot(evomodel, y = "s.num", sim.lines = TRUE,ylab="Susceptibles")```
```

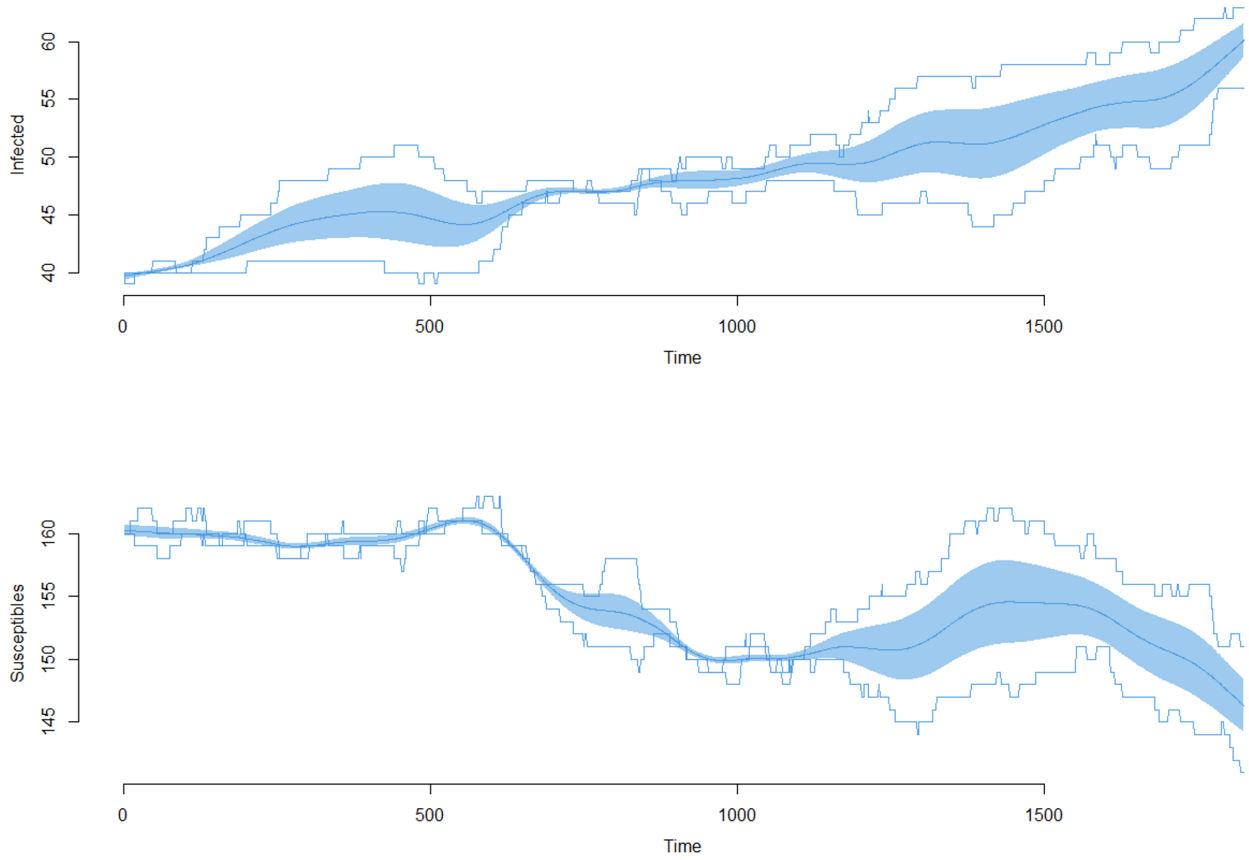


Figure 1: Fig3

To see additional time course data, type

```
evoplot(model=evomodel, name = "evonet_plots", path= "\path")
```

This command will automatically produce ~20 plots to the screen and a pdf titled "evonet_plots.pdf". In section IV, we give additional ways of seeing time course data.

III. Inputs

EvoNetHIV includes three types of parameters: standard epidemiological parameters, specialized parameters designed to take advantage the existing social-network functions given in the StatNet package, and the list of new or existing EvoNetHIV modules will be used in your runs. To minimize the number parameters sent to the *nw_setup* and *evorun* functions, we have placed all of three kinds of parameters into a single parameter list, *evoparams*.

a. Epidemiological parameters

In this section, we list of some of the more commonly manipulated epidemiological and printing parameters.

```
evoparams$initial_pop      = 10000
evoparams$initial_infected = 750
evoparams$n_steps          = 45*365
evoparams$nsims            = 16
evoparams$print_frequency = 365
```

For a complete list, we refer you to appendix 1. Parameters from a completed simulation can be viewed using commands like

```
evomodel$param[[1]]$initial_pop
evomodel$param[[1]]$initial_infected
```

b. Social network parameters

Our social network parameters are directly based off the StatNet package for social network dynamics. To simulate a simple "edges only" network (i.e., a network in each agent has the same mean degree and in which each agents partner with each other without regard to age or HIV status) for a population of size 1000 and mean degree (average number of partners at any given time) of 0.7, include the commands

```
initial_pop <- 1000
mean_degree <- 0.7
nw_form_terms <- "~edges"
target_stats <- initial_pop*mean_degree/2
```

Note here that StatNet counts edges that link two partners as two links, so we divide the mean degree by two in order to get the correct mean degree in StatNet. In section 9, we give examples of more complex sexual network structures that we have used in our studies.

c. Module list

As illustrated in the quick start guide in section, the modules that will be called by EvoNetHIV are specified using a command of the form:

```
modules <- c(
     "aging",
     "testing",
     "treatment",
     "viral_update",
```

```
        "coital_acts",
        "transmission",
        "evo_departures",
        "evo_arrivals",
        "summary_module")
```

Note here that each module is specified as string. EvoNetHIV translates these strings internally into the corresponding R functions given in EvoNetHIV.

In appendix 4 we a list of the built-in in functions are available to end users. To use one of these modules, simply add the corresponding text string to the module list. To see the exact commands used in each module, look for the corresponding file in https://github.com/EvoNetHIV/EvoNetHIV/tree/master/pkg/R

d. Creating your own modules

To create your own module, you will need to create and source a R function of the form

```
my_function <- function(dat, at) {
   # ... make some changes to the "dat" structure...
   return(dat)
}
```

and then add "my_function" to the list of modules in the module list above. In this template, "at" gives the timestep in days, and "dat" is a single, large data structure that includes the parameters, the agent attributes, the edgelist, and various summary statistics (described in more detail in section IV below). So, for example, if you wanted to specify that people under the age of 30 are twice as likely to use a condom, you could write

```
 Age_Based_Condom_Usage <- function(dat, at) {
   young_people <- which(dat$age < 30)
   dat$attr$condom_prob[young_people] <- 2*dat$param$condom_prob
   return(dat)
}
```

and add "Age_Based_Condom_Usage" to the modules list before coital acts.

IV. Outputs

EvoNetHIV produces three main outputs: time course data, agent attributes at the end of the simulation, and sexual partnerships at the end of the simulation.

a. Time Course data

Time course data is stored in a list called evomodel$epi where evomodel is the default name for the output given by evorun. EvoNetHIV stores time course data for 20-40 variables (listed in Appendix 2) depending on the application. Key variables of interest include:

```
evomodel$epi$timestep    #  Times population was sampled (set by evoparams$print_frequency).
evomodel$epi$susceptibles
evomodel$epi$prevalence
evomodel$epi$incidence   # Number of infections per year per 100 people
evomodel$epi$died_aids
evomodel$epi$mean_degree # Average number of links per person
```

Many of these are given in the default plots created by EvoNetHIV. To create your own plot, you may type something like

```
plot(evomodel$epi[[1]]$timestep, evomodel$epi[[1]]$prevalence)
```

Alternatively you can use the built-in EpiModel printing function

```
plot(evomodel, y = "prevalence", mean.line = FALSE, sim.lines = TRUE)
```

b. Agent attributes

Agent attributes are stored in a list called evomodel$attr[[i]] where i refers to the replicate number. EvoNetHIV stores roughly 70 attributes (listed in Appendix 3). Key attributes of interest include:

```
evomodel$attr[[1]]$Status   # 1 = Infected, 0 = susceptible, -1 = died of natural causes,
                            # -2 = died of AIDS
evomodel$attr[[1]]$Time_Inf # Time in years the agent got infected (fixed at infection)
evomodel$attr[[1]]$V        # Viral load (RNA copies /ml)
evomodel$attr[[1]]$CD4      # CD4 category; 1: CD4>500, 2: 350<CD4<500, 3: 200<CD4<350
                            # 4: CD4<200, 5: Died of AIDS
evomodel$attr[[1]]$age      # Age in years
evomodel$attr[[1]]$SPVL     # Setpoint viral load (fixed at time of infection)
```

These values can be graphed like any other list. To create a histogram of agent ages at the end of the simulation you could type

```
hist(evomodel$attr[[1]]$age)
```

To create a distributions of ages in replicate 2, you may type

```
hist(evomodel$attr[[2]]$age)      # Age in years
```

c. Sexual partners

Partnership data is stored in a structure called evomodel$el[[i]]. The edge list from the first replicate may be accessed by typing

```
evomodel$el[[1]][[1]]
       [,1] [,2]
  [1,]    3 1002   # Means agents 3 and 1002 were partnered at the end of the simulation
  [2,]    5 1325
  [3,]    8  656
  [4,]    8  658
  [5,]   11 1186
  ...
```

V. Case study – Effect of concurrency on HIV prevalence.

The model introduced in section II assumes everyone has the same number of connections. In actual populations, some people will have more partners than expected by chance. This can alter the rate of spread of HIV in populations. In the code given below, we use the concurrency functions in statnet to contrast complete monogamy from a model with concurrency.

Read in parameters, change default values for nsims, n_steps, popsumm_frequency, initial_pop, initial_infected. Use default modules.

```
#Read in default parameters (as list)
evoparams <- evonet_setup(
nsims=3,
popsumm_frequency =30,
 n_steps           = 365*30,
 initial_pop       = 300,
 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: concurrency model needs two target stats values: i) total edges and ii) total edges in concurrent relationship.

```
target_stats_monogamy <- c(0.35*evoparams$initial_pop, 0) #complete monomagy
target_stats_concurrency <- c(0.35*evoparams$initial_pop, 0.25*evoparams$initial_pop) #25% relationship
target_stats_list=list(target_stats_monogamy,target_stats_concurrency)
```

Create list to hold output of three model runs

```
output_list <- vector('list',length=length(target_stats_list))
```

Run model for each value of specified target stats

```
for(ii in 1:length(target_stats_list)){
  evoparams$target_stats <- target_stats_list[[ii]]
  nw <- nw_setup(evoparams)
  output_list[[ii]] <- evorun(modules,evoparams,nw)
}
```

save parameters and output to current working directory for future work if needed (e.g., getwd())

```
save(evoparams,output_list,file="concurrencny_Sim.RData")
```

Plot and compare prevalence time series for each target stat value (prevalence increases with increasing target stat # / mean degree values), line at 0.25 for reference

```
par(mfrow=c(2,2))
evoplot(output_list[[1]],variables = "prevalence",main="prevalence: monogamy")
abline(h=0.25)
evoplot(output_list[[2]],variables = "prevalence",main="prevalence: concurrency")
abline(h=0.25)
```

Appendix I – EvoNetHIV Parameters

EvoNetHIV includes approximate 400 parameters. In this appendix we organize these parameters to 12 different categories. Within the program, parameters are referenced using commands like

```
dat$param$initial_pop
```

where initial_pop is the total number of people at time 0. Upon completion, parameters can be seen using commands along the lines of

```
dat$param[[1]]$initial_pop
```

where the [[1]] indicates that you a viewing the value of parameters used in replicate one. To see all of the parameters type

```
dat$param[[1]]
```

The default values shown here can be changed using commands similar to those shown in section II. ## 1. Setup parameters

```
  initial_pop  = 100 #initial popn
  initial_infected  = 20
  n_steps      = 365*2
  model_sex    = "msm"
  model_name   = "evomodel"
  nsims        = 1
  ncores =1    # >1 when running on multi-core machine.
```

2. Printing options

```
  print_frequency = 10 # Print to output screen every 10 days.
  popsumm_frequency=1  # Frequency updating popsumm stats
  network_print_frequency = 100
  scrolling_output = TRUE
  QA_QC_pause_time = 3  # Time delay for viewing QA/QC stats
  plot_nw      = TRUE
  plot_mean_degree_by_age = FALSE # See plots of mean degree by age.
```

3. Sexual network terms (uses StatNet routines)

```
  nw_form_terms = "~edges + offset(nodematch('role', diff=TRUE, keep=1:2))",
  nw_coef_form  = c(-Inf, -Inf),
  target_stats  = initial_pop*0.7/2,
  relation_dur  = 50,
  d_rate =        3e-05, # default value in epimodel's netest
  nw_constraints = " ~.",
  dissolution="~offset(edges)",
  rm_offset_rel = F, # Remove same-role (MSM simulations) and same-sex
```

```
                    # (heterosexual simulations) relationships.
                    # If = T, function remove_offset_relationships
                    # will be called in initialize_module.
modes          = 1, # epimodel param, will change to 2 in
                    # "input_params_derived" if hetero model
```

4. Within-host dynamics of virus

```
VL_Function  = "aim2",  # Other option is "aim3" (aim 3 code)
vl_peak_agent_flag    = FALSE
max_time_inf_initial_pop = 365*2
V0                    = 1e-4
vl_peak_acute         = 7.7e6     # From Little et al 1999
vl_max_aids           = 2.4e6     # Piatak 1993
vl_increase_AIDS      = 1.0041122 # Slope of vl increase after aids onset;
                                  # 400-fold incr in 4 yrs
                                  # From O'Brien and Hendrickson (2013).
t_peak                = 21.0
t_acute               = 90
t_acute_phase2        = 32
acute_decline_phase2_rate = -.03
vl_decay_rate_phase2 = 0.02
prog_rate             = 0.14     # Gives ~0.5 Log incr in VL over 8-yrs
vl_exp_decline_tx     = -0.6
vl_full_supp           = 13.0    # Final VL after treatment complete
vl_undetectable        = 50.0
```

5. SPVL variation

```
AverageLogSP0         = 4.5
VarianceLogSP0        = 0.8
Heritability          = 0.36     # Heritability of SPVL.
max_spvl_allowed      = 7.0
min_spvl_allowed      = 2.0
max_vl_viralcont_spvl = 7
min_vl_viralcont_spvl = 2.5
MutationVariance      = 0.01
```

6. Transmission terms

```
transmission_model    = "hughes"  # Relat betw VL and trans prob. Other choice: "hill"

# Relevant for the "hill" function for the relationship between VL and transmission
InfRateBaseline       = 0.0000003 # 0.0001178/365     # Lingappa 2010
InfRateExponent       = 3.52      # Lingappa 2010
MaxInfRate            = 0.002     # Asymptotic function from Fraser 2007
                                  # From: P_inf(1 year) = 1 - (1 - P_inf(1 day))^365
```

```
VHalfMaxInfRate       = 13938     # Fraser 2007
HillCoeffInfRate      = 1.02      # Fraser 2007
shape_parameter       = 3.46      # 3.46 is from Fraser
Dmax                  = 9271      # From Fraser, max time (days) asymptomatic state
D50                   = 3058      # From Fraser, spVL at which duration is half maximum
Dk                    = 0.41      # From Fraser, Hill coefficient
Flat_Viral_Load       = 0         # If 1,  VL will the same during acute period
                                  # Added for counter-factual hypothesis

# Relevant for the "hughes" for the relationship between VL and transmission
trans_lambda          = 0.000247   # From Jim Hughes
trans_lambda_init     = 0.003      # See Hughes et al. (2012, JID) for details
trans_lambda_alpha    = 5

# Relevant for both transmission models
trans_RR_uses_condoms= 0.22    # From Hughes et al. (2012, JID) Table 4
trans_RR_LogV         = 2.89    # From Hughes et al. (2012, JID) Table 4
trans_VLbase          = 4.0     # From Jim Hughes
trans_RR_circumcised = 0.53    # From Hughes et al. (2012, JID) Table 4
trans_RR_age          = 0.67    # Older people at lower risk (Note: 0.82 in paper,
                                # 0.67 from Jim Hughes)
max_age_RR_age        = 45      # Threshold age, where ages above threshold have same RR
trans_base_age        = 35,     # From Jim Hughes
trans_RR_STI          = 2.7     # placeholder for generic sti
trans_RR_insertive_anal_msm = 2.9  # From Patel 2014, but adjusted for circumcision.
trans_RR_receptive_anal_msm = 17.3 # From Patel 2014, but adjusted for circumcision.
trans_RR_ins_ai       = 2.9
trans_RR_rec_ai       = 17
trans_RR_acute_phase = 1.0     # Increased infectiousness during acute phase
trans_RR_receptive_vaginal=1   # 1 per Jim Hughes ref; RR=2 if using data from Patel et al
trans_RR_vaccine      = 0.01
perc_virus_vaccine_sens = 0.99

susceptibility_var = 0.0       # First added for AgeAndSPVL work 3/2019 by SMG
```

8. Demographic parameters

```
min_age                  = 18,
max_age                  = 55,
age_new_adds              = "min_age", # "mixed" (some min_age, others older)
                                       # or "linear_decline_18_55"
prop_new_agents_min_age   = 0.45, #for "mixed" see above line
asmr_data_male            = "usa_men_18_to_100",#other option: "south_africa_male"
asmr_data_female          = "south_africa_female",
initial_agedata_male      = "usa_men_18_to_100", #other options:"south_africa_male_16_to_100_2014", "
initial_agedata_female    = "south_africa_female_16_to_100_2014"
                            # other options: "stable_age_no_hiv_dist"
aids_death_model         = "cd4",        # c("Gamma_Death","daily_prob","cd4")
death_rate_constant      = 0.000003,       # 0.000003 is from CASCADE
death_rate_exponent      = 6.45,         # 6.45 is from CASCADE
cd4_cat1_death_prob      = 0.0000112,    # Prob. of death for cd4 cat1
```

```
 cd4_cat2_death_prob      = 0.0000148,    # Prob  of death for cd4 cat2
 cd4_cat3_death_prob      = 0.0000333,    # Prob of death for cd4 cat3
 cd4_cat4_treated_death_prob = 0.0000760, # Prob death for cd4 cat4(aids) on tx
 cd4_prob_incr_nadir      = 0.03,         # Prob of improving one cd4 cat from nadir
 cd4_prob_incr_nadir_minus= 0.0005,       # Prob of improving one cd4 cat from nadir -1
 time_in_aids             = 475,
 birth_model              = "poisson_birth_numbers",
                             # Other options: "births=deaths", "constant_number", "exponential_growth"
                             # "poisson_birth_numbers", "exponential_growth","constant_rate",
baseline_input_exp_growth = 0.007, # Used with "exponential_growth"  MUST BE SCALED BY HAND
 birth_model              = "poisson_birth_numbers", # Other options: "births=deaths",
                            # "poisson_birth_numbers", "constant_rate", "constant_number"
 contstant_birth_number   = 0,
 constant_birth_rate      = 0.0001306,
 poisson_birth_lambda     = NA,
 poisson_birth_base       = 0.01370, # Scaled to init pop size in 'input_derived_parameters',
 pop_growth_rate_annual   = 0.01,  # as proportion, x100 for percent
 constant_rate_spread_out = 0.01,   # Birth model: "constant_rate_spread_out"
 births_per_year          = 1,     # Birth model: "constant_number_spread_out"
```

9. Sexual behavior parameters

```
 mean_sex_acts_day        = 0.2
 aids_sex_cutoff_prop     = 0.47   # Prop time-in-aids after  coital acts cease; Holllingsworth 2008
 prob_sex_by_age          = FALSE  # Does the prob sex (given that one is in a relationship) decrease
 prob_sex_age_19          = 0.285  # When prob_sex_by_age == TRUE (then ignore mean_sex_acts_per_day)
 max_age_sex              = 70     # When prob_sex_by_age == TRUE (linear decline with age from age 19

 # Parmeters related to disclosure and condom use
 disclosure_prob          = 0.9,    # PUMA
 act_redux_discl          = 0.0,    # MARDHAM
 condom_prob              = 0.5,
 condom_prob_sd           = 0.0373, # Standard deviation for condom prob from MARDHAM simulation
 condom_prob_change       = F,      # For condom_prob to be 0 initially, then increase
 condom_prob_max          = 0.5,    # Used in hill function if condom_prob_change == T
 condom_prob_inflect      = 365*12, # Used in hill function if condom_prob_change == T
 condom_prob_pow          = 4.1,    # Used in hill function if condom_prob_change == T
 RR_cond_male_concurrent  = 1.438,
 RR_cond_fem_concurrent   = 1.0,
 condom_use_rel_dur = FALSE,
 condom_use_age = FALSE,
 age_condom_use_halves = 50, # Only used when condom_use_age is true

 percent_condom_users= 1, # Non-users never use condoms w/ other non-users.
                          # (users use condoms with some prob)
 condom_use_rel_dur = FALSE,
 condom_use_age = FALSE,
 age_condom_use_halves = 50,   # Only used when condom_use_age is true
 individual_condom_prob   = F, # Set to T for condom probability to be at the agent level,
 individual_condom_prob_compromise_method = "mean", # Others: "max" "min"
                                     # method for 2 agents with diff condom use patterns
```

```
 individual_condom_prob_var = 0.5, # Agent level condom probability,
 individual_condom_prob_rc = 0.5,  # Agent level condom prob with PrEP and risk compensation
                                   # No risk compensation by default

# STI/circumcision probabilites for agents (used in "vital new additions" fxn)
 circum_prob               = 0.85
 sti_prob                  = 0.0 # Used in "vital_new_additions"

# Miscellaneous
 sti_prob_att              = NA,
 circum_prob_chg           = c(0.45,   0.5,    0.55,   0.6,    0.65,   0.7,    0.75,   0.85)
 circum_prob_yr_chg        = NA # c(12*365, 13*365, 15*365, 16*365, 18*365, 20*365, 22*365, 24*365),
 prop_AI                   = 0.10
 mean_prop_acts_AI         = 0.4266
 sd_prop_acts_AI           = 0.2146
# MSM roles: "social_role_msm"
 role_props                = c('I'=0.24, 'R'=0.27, 'V'=0.49)
                             # To activate msm roles, set role props to
                             # c("I"=x,"R"=y,"V"=z), where x+y+z=1 and are >=0
 role_trans_mat           = matrix(c(1,0,0,0,1,0,0,0,1),
                                   nrow=3,dimnames=list(c("I","R","V")))
 prob_iev                  = 0.4     # Average of Mardham and PUMA
```

10. Vaccination parameters

```
   start_vacc_campaign      = 5e5
   perc_vaccinated          = 0.5
   target_vacc_att          = FALSE
   vacc_eff_duration        =  365*3,    risk_comp_cond           = F   # Set to T to reduce condom us
                                 # and vaccinated, infected, undiagnosed individuals.
   risk_comp_cond_rr        = 0.70,
   risk_comp_degree         = F    # Set to T to reduce  condom use in vaccinated susceptibles
                                   # and vaccinated, infected, undiagnosed individuals.
 risk_comp_degree_rr      = 1.3
```

11. Testing parameters

```
   testing_model            = "interval",
   mean_test_interval_male  = 365,
   mean_test_interval_female = 442,
   mean_test_interval_under25 = 365,
   test_result_delay = 35, # Delay betw infection andantibodies for diagnosis
   reduction_test_interval_enhanced = 0.1,  # Reduction in test interval for "enhanced" testing indiv
   prob_enhanced_testing_before_campaign = 0.3, # % tested more frequently before scale-up campaign
   prob_enhanced_testing_after_campaign = 0.7,  # % tested more frequently after scale-up campaign
   testing_limit = "percent_agents", # For models in which there is an upper limit to number tested.
                                     # Other is "percent_agents_minus_diagnosed"
```

12. Treatment parameters

```
  prob_care                = 1.0,   # Prob of eligible for care before "under care" campaign
  prob_eligible_ART        = 1.0
  prob_eligible_2nd_line_ART = 1.0
  second_line_elig          = "vl_only"
  prob_tx_droput           = 0
  compact_el_divisor=1e5
  ave_rel_dur_start = 5*365
  tx_type                = NA
  mean_trtmnt_delay = 1
  mean_time_tx = 0  # used when tx_type = "time_dist"
  sd_time_tx = 0    # used when tx_type = "time_dist"
  start_treatment_campaign = 5e5
  tx_limit = "absolute_num" # used with social_treatment_module_multiple_criteria.
  max_num_treated = 5e5
  max_num_treated_begin = 5e5 # Num during pre-campaign (for social_treatment_module_multiple_criteri
  max_num_treated = 5e5       # Num during big campaign (for social_treatment_module_multiple_criteri
  proportion_treated       = 1
  yearly_incr_tx         = 0 # Setting this to 0.1 would mean 10% more people get treated each year
  proportion_treated_begin = 0.0
  start_treat_before_big_campaign = 5e5
  num_randomly_chosen_start_campaign = 0  # Num outside the prioritized group who get treated
   num_treated_start_campaign = 0  # Total num treated at the start of the campaign
  num_enhanced_start_campaign = 0 # Num diag or receiving enhanced testing
 # Below four parameters are used in social_treatment_sex_age module
      cov_prob      <- c(0.0, 0.01, 0.021, 0.030, 0.049, 0.100, 0.191, 0.283, 0.402, 0.78) # Coverage
      cov_prob_yrs  <- c(0, 11:18, 23) # Years at which coverage level changes
      cov_prob_scal <- matrix(c(0.7242, 0.9955, 1.2593, 0.5895, 0.8103, 1.0251), ncol = 2, dimnames =
                      # Values are by sex, then by age within sex
      cov_prob_ageg <- list(c(15, 25), c(25, 35), c(35, 56))

  # -- additional treatment parameters --- #
  under_25_flag           =  FALSE
  tx_in_acute_phase        = FALSE
  tx_schedule_props        =c("F"=1,"V"=0,"N"=0,"P"=0)
  treatment_threshold      = 1e4    #VL raw,not log10 transformed
  cd4_treatment_threshold  = 0
  cd4_trt_guidelines_chgs  = NA # Default for SA eligibility changes: list(4, 3:4, 2:4, 1:4)
  attr_treatment_threshold  = 5  # Under development.  When finished, this parameter will
  # allow one to target therapy to attribute groups
  # "attr_treatment_threshold" and higher (e.g. target therapy
  #  to people with the most connnections)
  min_inf_time_for_treat   = 0     # number of timesteps (days)
  treat_thresh_partners    = 0
  max_treated              = NA  #relevant for "social_treatment_john" module
  #this value is set within module at start of tx campaign,
  #NA value is flat that indicates it hasn't been initialized yet
```

13. PrEP parameters

```
no_past_partners_time_prep = 365
min_past_partners_prep = 1
min_current_partners_prep = 1
min_pos_partner_duration = 50
prep_recent_test = 183
percent_eligible_on_prep = 1
start_prep_campaign = 5e5
```

14. Flags for saving data

```
save_network  = FALSE
save_coital_acts = FALSE
save_vl_list = FALSE    #TRUE to graph individual agent VL
save_infection_matrix =FALSE
save_partner_list = FALSE
save_trans_probs = FALSE # Used in "trans main module"
                         # Saves timestep/susc. ids/ trans. probs in "trans_probs_list"
Save_VL_Histories  = FALSE
```

15. Flags for running on high performance computer cluster

```
hpc          = FALSE # Running on the UW's high performance computer (hyak)
hyak_par     = FALSE # Running on hyak and parallelized run
```

16. Parameters for multi-locus drug resistance model

```
Max_Allowable_Loci = 5
prob_CYP_6_slow  = 0
step_size_C    = 0.0004 # Step size in C program simulator
s_CD4       = 10        # Rate of input of CD4 T-cells from the thymus
m_CD4       = 0.01      # Natural death rate of CD4 T-cells
k           = 0.0000001 # Rate at which viruses kill off CD4 T-cells
r_inf_cells = 1.8       # Growth rate of infected cells at start of infection
r_r_inf_cells  = 1.3    # Growth rate of drug resistant cells start of infection
d_inf_cells  = 0.6      # Death rate of infected cells
f_M         = 0.02      # Frac of target cells become moderately long-lived inf cells
f_L         = 1e-6      # Frac of target cells  become very long-lived infected cells
d_M         = 0.04      # Death rate of moderately long-lived infected cells
d_L         = 0.001     # Death rate of latently infected cells
p_inf_cells = 1000      # Rate actively infected cells produce virus
p_M         = 100       # Rate moderately long-lived infected cells produce virus
p_L         = 10        # Rate latently infected cells produce virus
M_act       = 0.01      # Rate  M cells convert into productively infected cells I
L_act       = 0.0005    # Rate latently infected cells become productively inf cells
```

```
c           = 50        # Clearance rate of free virus
mu          = 3e-5      # Rate at which viruses mutate from sensitive to resistant
TransBottleneck = 3e8   # Reduction in viral load associated with transmission
no_loci     = 5         # Number of loci conferring drug resistance
cost1       = 0.05      # Fitness cost drug resistance mutation at locus 1
cost2       = 0.05      # Fitness cost drug resistance mutation at locus 2
cost3       = 0.05      # Fitness cost drug resistance mutation at locus 3
cost4       = 0.05      # Fitness cost drug resistance mutation at locus 4
cost5       = 0.05      # Fitness cost drug resistance mutation at locus 5
                        # (note: this mut negates cost at locus 2)
cost_reduct4on2 = 0.9   # Extent to which mutation 4 mitigates the cost of mutation 2
cost_reduct5on1 = 0.5   # Extent to which mutation 5 mitigates the cost of mutation 1

additive_fitness = 0    # 1: fitness = 1 - cost1 - cost2 -... - cost5.  <>
                        # 0: multiplicative: Fitness = (1-cost1)*(1-cost2)*...(1-cost5)

DrugDose1 =  200, DrugDose2 = 200, DrugDose3 = 200.0,  DrugDose4 = 200.0, # Dose of drugs taken
drug_decay1  = 1.0      # Per day clearance rate of drug 1
drug_decay2  = 1.0      # Per day clearance rate of drug 2
drug_decay3  = 1.0      # Per day clearance rate of drug 3
drug_decay4  = 1.0      # Per day clearance rate of drug 3

drug_2nd_decay1 = 0.1   # Second-phase decay rate drug 1
drug_2nd_decay2 = 0.1   # Second-phase decay rate drug 2
drug_2nd_decay3 = 0.1   # Second-phase decay rate drug 3
drug_2nd_decay4 = 0.1   # Second-phase decay rate drug 4

conc_2nd_phase1   = 1.0e-70  # Drug conc second-phase starts for drug 1
conc_2nd_phase2   = 1.0e-70  # Drug conc second-phase starts for drug 2
conc_2nd_phase3   = 1.0e-70  # Drug conc second-phase starts for drug 3
conc_2nd_phase4   = 1.0e-70  # Drug conc second-phase starts for drug 4

min_adherence1 = 0   # Each agent has an randomly adherence level betw min and max
max_adherence1 = 1
min_adherence2 = 0
max_adherence2 = 1
min_adherence3 = 0
max_adherence3 = 1
min_adherence4 = 0    .
max_adherence4 = 1

adherence_type = 1:2 #1=random, 2=cyclic
adherence_type_prob=c(1,0) #default: all agents adherence type 1,
adherence_days_high=5 #for cyclic adherence
adherence_days_low=2 #for cyclic adherence
aherence_days_high_prob=0.9 #for cyclic adherence
aherence_days_low_prob=0.1 #for cyclic adherence

BaseIC50Drug1 = 200.0, BaseIC50Drug2 = 200.0, BaseIC50Drug3 = 200.0,  # Concentration of drug that
BaseIC50Drug4 = 2.0   # Drug 4 is some super-effective 2nd-line therapy combo

Interaction_Model_Drugs12 = 1  # 1 = Bliss independence, 2 = Simple saturation (Huang et al. 2003),
                               # 3 = Lowe additivity (not yet implemented)
```

```
                                # Background: Current params assume that drugs 1 and 2 are both NRTI
                                # Therefore, they  compete for the active site, reducing  inhibition
                                # This implements different math ideas for how they  combine

FC_D1_Mut1    = 50.0    # Effect of mut 1 on the IC50  of drug 1. (Fold-change from baseline)
FC_D1_Mut2    = 1.0
FC_D1_Mut3    = 1.0
FC_D1_Mut4    = 1.0
FC_D1_Mut5    = 1.0

FC_D2_Mut1    = 1.0
FC_D2_Mut2    = 50.0
FC_D2_Mut3    = 1.0
FC_D2_Mut4    = 1.0
FC_D2_Mut5    = 1.0

FC_D3_Mut1    = 1.0
FC_D3_Mut2    = 1.0
FC_D3_Mut3    = 10.0
FC_D3_Mut4    = 5.0
FC_D3_Mut5    = 1.0

FC_D4_Mut1    = 1.0
FC_D4_Mut2    = 1.0
FC_D4_Mut3    = 1.0
FC_D4_Mut4    = 1.0
FC_D4_Mut5    = 1.0

AbsoluteCut   = 1.0e-7   # Density of cells (or viruses) that correspond to 1 cell per body.
StochasticCut = 1.0e-6   # Density below which changes occur stochastically

Dosing_Interval =1  # 1: Once daily dosing, 2: Twice daily dosing
Therapy_Type = 1  # 1: three individual pills, 2: drugs 1 and 2 are contained  a single pill
                  # 3: All three drugs are contained within a single pill

# Stockout parameters (e.g., no gets Drug 1 btw StopDrug1 and RestartDrug1)
# These apply to all patients regardless of their adherence
StopDrug1    = 1000000000.0  # Stock out time (in days) for drug 1
RestartDrug1 = 1000000000.0  # Drug 1 becomes available again
StopDrug2    = 1000000000.0  # Stock out time (in days) for drug 2
RestartDrug2 = 1000000000.0  # Drug 2 becomes available once again
StopDrug3    = 1000000000.0  # Stock out (in days) for drug 3
RestartDrug3 = 1000000000.0  # Drug 3 becomes available once again
StopDrug4    = 1000000000.0  # Stock out (in days) for drug 3
RestartDrug4 = 1000000000.0  # Drug 3 becomes available once again
```

Appendix 2 – Agent attributes

Within the program, parameters will be referenced using commands like

```
dat$attr$id
```

where id is a unique identifier for each agent. After the program has completed, the final values for each attribute can be retrieved using commands like

```
dat$attr[[1]]$V
```

where V stands for viral load and the 1 inside the brackets refers to the first replicate. The complete list of attributes is:

Basic epidemiological attributes

id
Status
arrival_time # Time that agent entered the population age
age_infection
sex
role
V
CD4
Time_Death susceptibility
Time_Inf # Time that infected agent got infected Time_Inf_Adj # Time_Inf plus days under treatment (for CD4 dynamics) Generation
circum # 1 if circumcised, 0 if not condom_user
individual_condom_prob
sti_status
ai_prob
insert_quotient total_acts
total_disc_acts

Attributes related to risk group membership

att1 # 1 = high risk group, 2 = low risk group

Attributes related to viral dynamics and set point viral load

r0
vl_peak_agent
d_acute
vl_phase2_trans
rate_phase2
SetPoint
LogSetPoint
ViralContribToLogSP0
EnvirContribToLogSP0
RandomTimeToAIDS

Attributes related to CD4 T-cell counts

CD4_time
CD4_initial_value
CD4_treatment_delay_index
spvl_cat
CD4_time_death
CD4_nadir
CD4_at_trtmnt
num_consec_VL_gt1k
start_aids_cd4
start_max_aids

Attributes describing the agent's infector (donor) and infectees (recipients)

Donors_V
Donors_treated
Donors_treated_2nd_line
Donors_CD4
Donors_ViralContribToLogSP0
Donors_EnvirContribToLogSP0
Donors_Total_Time_Inf_At_Trans
Donors_Generation
Donors_Index
Donors_age
Donors_LogSetPoint
Donors_SetPoint
Donors_d_acute

Donors_diag_status
NumRecipients

Attributes related to the cascade of care

eligible_care
eligible_ART
eligible_vl_test

Attributes related to HIV testing, diagnosis, and disclosure

vl_at_test
cd4_at_test
eligible_2nd_line_ART
treated_2nd_line
diag_resist_status
diag_resist_time
last_neg_resist_test
time_init_2nd_line
last_neg_test
diag_status
diag_time
disclosure_status

enhanced_testing
time_hiv_sex_act
time_hiv_sex
last_disc_sex
rand_prob_test
ever_enhanced_testing
rand_prob_test_init

Attributes related to treatment

treated
tx_init_time
tx_schedule
tx_stop_time
min_time_tx
tx_dropout

Drug adherence attributes

adherence_start
adherence_type
Adherence1
Adherence2
Adherence3
Adherence4

Attributes used by the pre-exposure prophylaxis module

pos_partner_duration
no_partners_past_prep
no_partners_now_prep
have_diag_partner
have_disclosed_partner
on_prep
prep_decrease
eligible_for_prep
eligible_for_prep_1
eligible_for_prep_2
prep_init_time
prep_discontinue_time
prep_list
known_pos_partner_duration
partner_recent_test
have_suppressed_partner
have_unknown_status_partner
agent_condom_user
last_ts_relationship
last_ts_multiple_relationships

Attributes used by the vaccination model

vaccinated
vacc_init_time
vacc_eff
vacc_rr
virus_sens_vacc
virus_sens_drug
virus_part_res_drug
trial_status
LogSetPoint_genotype
vacc_status_at_inf
phi
mu
sigma
theta
m
vaccination.dates.stack

Attributes used by the multi-locus drug resistance module

Drug1
Drug2
Drug3
Drug4
K
Imm_Trig
ChronPhase
OnDrug
CYP_6_slow
aim3_no_muts
Aim3RoundingErrors
aim3_mutations_long
virus_1_plus_drug_muts
virus_3_plus_drug_muts

```
\newpage
# Appendix 3 -- Time course data

Within the program, time course results are stored using commands like

‘‘‘r
dat$epi$num_infected
```

where dat*epi*num_infected is a list that gives the number of infected people every dat*param*epi$frequency step. After the program has completed, the final values can be retrieved using commands like

```
dat$epi$num_infected
```

where num_infected and the 1 inside the brackets refers to the first replicate. Note that unlike the dat*popanddat*attr that we don't use the [[i]] brackets for epi (check)! The major time course variables are:

```
# Key epidemic statistics
timestep
num
s.num
i.num
alive
prevalence
new_infections
mean_trans_prob
new_diagnoses
susceptibles
total_infections_alive
total_infections_not_treated
births
aids_deaths
natural_deaths
aged_out


# CD4 and AIDS
cd4_gt_350
cd4_200_350
cd4_0_200
no_in_aids_gamma
no_in_aids_cd4


# Circumcision and treatment
circum_prev
no_treated
percent_suppressed
no_treated_undetectable
mean_vl_pop_untreated
percent_treated_undetectable
total_pills_taken
mean_degree_inf_treated


# Social network statistics
no_edges
mean_degree
mean_degree_inf_untreated
mean_degree_male
prop_nodes_degree_0
prop_nodes_degree_1
prop_nodes_concurrent


# Setpoint viral load statistics
mean_spvl_pop_all
mean_vl_pop_all
mean_spvl_incident
mean_spvl_pop_untreated
```

```
# Statistics related to donors
percent_donor_acute
mean_time_donor_infected_incident


# Key statistics broken out by risk group
generic_att_percent_cat_1
generic_att_percent_cat_2
generic_att_percent_inf_cat_1
generic_att_percent_inf_cat_2
generic_att_mean_degree_cat_1
generic_att_mean_degree_cat_2
generic_att_percent_vacc_cat_1
generic_att_percent_vacc_cat_2
prevalence_attr_1
prevalence_attr_2
new_infections_attr_1
new_infections_attr_2
susceptibles_attr_1
susceptibles_attr_2
mean_spvl_incident_attr_1
mean_spvl_incident_attr_2
prop_on_prep_attr_1
prop_on_prep_attr_2
prop_eligible_prep_attr_1
prop_eligible_prep_attr_2


# Key statistics broken out by age and sex
alive_female
alive_male
inf_men
inf_women
prev_15to24
prev_15to49
prev_f_15to24
prev_f_15to49
prev_m_15to24
prev_m_15to49
inf_under30
inf_30to50
inf_over50
mean_age_incident
mean_age_died_AIDS
mean_age_infecteds
mean_age_susceptibles
mean_degree_under_30
mean_degree_30_50
mean_degree_over_50
mean_degree_female
treated_inf_men
treated_inf_women
```

```
treated_inf_under30
treated_inf_30to50
treated_inf_over50
```

Appendix 4 – EvoNetHIV Modules

As described in section 4, EvoNet modules are invoked by putting the corresponding string into the module list in the run script. Within the program, modules are defined using commands of the form

```
my_function <- function(dat, at) {
   # ... make some changes to the "dat" structure...
   return(dat)
}
```

In the following, we give a very brief overview of the most commonly used module. Most of these have multiple inputs and outputs. For simplicity, we focus here on the major inputs and putputs.

Default modules

aging

Increases ages for all agents by one day.

testing

Periodically tests agents to see if they are HIV+. HIV+ agents who get tested are

treatment

Determines which diagnosed agents get treated. Has options to treat agents at random or according to CD4 count. If treated set treated to 1.

viral_update

Updates viral load and CD4 counts. If treated, viral load will decline and CD4 counts will slowly increase.

coital_acts

Determines which agents have sex. Stores list of serodiscordant agents that will be processed by the transmission module

transmission

From a list of serodiscordant couples who had sex, determine which agents get infected. If an agent gets infected, set status to 1.

evo_departures

Determines which agents have died or aged out of the model. If the agent died of AIDS, set status to -2. If the agent died of natural causes, set status to -1

evo_arrivals

Determines which agents have entered the model. New agents are typically assumed to of age 16.

summary_module

Updates time course dataset (dat$epi) for subsequent graphing

Speciality modules used in our targeted treatment-as-prevention paper

targeted_treatment2

Similar to the treatment module, but includes a much longer list of options for targeting specific groups of people for linkage to care. To use this version, replace "treatment" with "targeted_treatment2" in the modules list.

social_testing_diagnosis_eligibles_only_module

Similar to testing, but limits testing to those who are "eligible for care" (i.e., people who are connected enough to the health care system to receive an HIV test).

adherence

Allows for agents to vary in their likelihood of taking antiretroviral drugs.

treatment_dropout

Allows for agents to stop treatment

viral_update_delayed_rebound

Replaces viral_update in runs that include the "treatment_dropout" module. This module accounts for observations that there usually a delay between halting treatment and seeing a rebound in viral load.

cd4_update2

Updates CD4 counts. Usually used in connection with targeted_treatment2 and treatment_dropout ### viral_update_cd4_daily Similar to