London MSM simulated epidemic history

Stephane

3/27/2017

Epidemic simulation

The goal is to replicate a simple transmission history in a population that is comparable to MSM in London. The simulation is based on a system of ordinary differential equations which describe the dynamics of the number of infected hosts in different categories. The model allows to vary how individual characteristics influence transmission to test methods of estimation of transmission risk under different scenarios departing from baseline scenario shown here.

A single epidemic simulation is produced by the R script `model0.r`. Here, we read the source code, load necessary packages and execute it chunk by chunk

```
library(knitr)
read_chunk('model0.R')
```

```
require(phydynR)
require(deSolve)
require(Rcpp)
```

Define input parameters

The epidemic history is modeled with a system of ordinary differential equations determining transmission and transition through 5 infection stages (early HIV infection (EHI) , 3 chronic stages based on CD4 and AIDS), 4 age groups (based on quartiles of observed diagnosed individuals in UKDRDB) and 3 diagnosis states (undiagnosed, diagnosed untreated and diagnosed under treatment). Individuals are further stratified by an arbitrary binary risk characteristic influencing transmission. The population is thus structured in 120 states or demes. In addition, we model importation of infections into London by adding a *source* compartment that represents infected hosts outside of London MSM group. The source compartment is parametrized by its initial size, migration rate and growth rate.

```
##- Age progression
##(by quantiles: 18.0, 27.0, 33.0, 40.0, 80.5)
age_rates <- c(agerate1 = 1/9/365
    , agerate2 = 1/6/365
    , agerate3 = 1/7/365
    , agerate4 = 1/40.5/365
)

##- Natural history (from Cori et al. AIDS 2015)
stage_prog_yrs <- c( .5, 3.32, 2.7, 5.50, 5.06 )
stageprog_rates <- setNames( 1 / (stage_prog_yrs  * 365 )
 , c('gamma1', 'gamma2', 'gamma3', 'gamma4', 'gamma5')  )

pstarts <- c( pstartstage1 = 0 #NA
 , pstartstage2 = 0.76
 , pstartstage3 = 0.19
 , pstartstage4 = 0.05
 , pstartstage5 = 0
)
```

```
theta <- c( age_assort_factor = .5 # power of age difference
  , pRiskLevel1 = .8 # proportion in low risk group
  , srcMigrationRate = 1/50/365 # per lineage rate of migration to source
  , srcGrowthRate = 1 / 3 / 365 #
  , src0 = 1e3  # initial source size
  , inc_scale = .03 # based on docking (see below) # initial = .03 # 0.09401734
  , max_diag_rate =  1/3 # based on docking (see below) # initial = 1/3 (time 2 diag of 3 yrs) # 0.6622
  , diag_rate_85 = 1/10
  , accel_diag_rate = 1/7 # based on docking (see below ) # initial = 1/7 # accel of logistic function
  , treatmentEffectiveness = .95 # slows stage progression
  , pstarts
    , age_rates
    , stageprog_rates
)

theta_default <- theta
```

Transmission parameters

Age groups are based on quantiles of observed age distribution of MSM diagnosed with HIV in London ("HIV and STIs in Men Who Have Sex with Men in London" 2014) and transmission rates are independent of age. However, in baseline scenario, transmission is allowed to vary according to *weights* provided by risk category, stage of infection, treatment status and according to age assortativity:

- A proportion of 20% of the population are deemed to be at high risk with a ten-fold increase in transmission relative to low risk counterparts.
- We let infection stage influence transmission rates with a ten-fold increase in probability of transmission in early HIV infection and a three-fold increase in AIDS stage. Mean time of progressions to CD4 stages ($\gamma_i, i = 1, \ldots, 5$) and proportion in each CD4 category after seroconversion ($\pi_i, i = 1, \ldots, 4$) were obtained from Cori et al.

- Relative to undiagnosed individuals, diagnosed and treated patients have a reduction in transmission by respectively a factor 2 and 20 .
- An age assortativity parameter is introduced in the transmission matrix which causes transmission rates to decrease as a power law function of the difference in age.

```
print(theta)
```

```
##      age_assort_factor            pRiskLevel1       srcMigrationRate
##           5.000000e-01           8.000000e-01           5.479452e-05
##          srcGrowthRate                   src0              inc_scale
##           9.132420e-04           1.000000e+03           3.000000e-02
##          max_diag_rate           diag_rate_85        accel_diag_rate
##           3.333333e-01           1.000000e-01           1.428571e-01
## treatmentEffectiveness           pstartstage1           pstartstage2
##           9.500000e-01           0.000000e+00           7.600000e-01
##           pstartstage3           pstartstage4           pstartstage5
##           1.900000e-01           5.000000e-02           0.000000e+00
##               agerate1               agerate2               agerate3
##           3.044140e-04           4.566210e-04           3.913894e-04
##               agerate4                 gamma1                 gamma2
##           6.764756e-05           5.479452e-03           8.252187e-04
##                 gamma3                 gamma4                 gamma5
```

```
##           1.014713e-03           4.981320e-04           5.414478e-04
```

```
##- transmission by stage
nh_wtransm <- c(
    nh1 = 1
    ,nh2 = .1
    ,nh3 = .1
    ,nh4 = .1
    ,nh5 = .3
)
##- transmission by age
age_wtransm <- c(
    age1 = 1
    , age2 = 1
    , age3 = 1
    , age4 = 1
)
##- transmission by treatment status (undiag, diag, treated)
care_wtransm <- c(
    care1 = 1
    , care2 = .5
    , care3 = .05
)
##- transmission by risk group
risk_wtransm <- c(
    risk1 = 1
    , risk2 = 10
)


## time axes & funcs
time_res <-  52 * (2013 - 1979 )  # time steps / week
year0 <- 1979
year1 <- 2013
date0 <- as.Date('1979-01-01')
date1 <- as.Date('2012-12-31')
times0 <- 0
times1 <- as.numeric( date1 - date0 )
times_year <- seq(year0, 2013, length.out = time_res) #to end of 2012
times_day <- seq( 0, times1, length.out = time_res )
days2years <- function( d ){
    year0  + (year1 - year0) * d / (times1 - 0 )
}
years2days <- function(y)
{
    (times1 - times0) * (y - year0) / (year1 - year0)
}
```

```
## list of compartments
N_NH_COMPS <- 5
N_AGE_COMPS <- 4
N_RISK_COMPS <- 2
N_CARE_COMPS <- 3
#~ also remember source
```

```
NH_COMPS <- paste(sep='', 'stage', 1:N_NH_COMPS )
AGE_COMPS <- paste(sep='', 'age', 1:N_AGE_COMPS )
RISK_COMPS <- paste( sep='', 'riskLevel', 1:N_RISK_COMPS )
CARE_COMPS <- paste(sep='', 'care', 1:N_CARE_COMPS)

COMPS_list <- list( NH_COMPS, AGE_COMPS, CARE_COMPS, RISK_COMPS )

NH_COORDS <- list()
AGE_COORDS <- list()
CARE_COORDS <- list()
RISK_COORDS <- list()
DEMES <-c()
k <- 1
for ( nh in NH_COMPS ){
    for (age in AGE_COMPS){
        for (care in CARE_COMPS){
            for (risk in RISK_COMPS){
                NH_COORDS[[nh]] <- c( NH_COORDS[[nh]] , k )
                AGE_COORDS[[age]] <- c( AGE_COORDS[[age]], k )
                CARE_COORDS[[care]] <- c( CARE_COORDS[[care]], k )
                RISK_COORDS[[risk]] <- c( RISK_COORDS[[risk]], k )
                DEMES <- c( DEMES, paste(sep='.', nh ,age, care, risk ))
                k <- k + 1
            }
        }
    }
}
DEMES <- c( DEMES, 'src' )
m <- length(DEMES)

# indicators for each deme; note C-indexing
NH = rep(NA, m)
AGE = rep(NA, m)
CARE = rep(NA, m )
RISK = rep(NA, m)

k <- 1
for ( care in CARE_COMPS ){
    CARE[ CARE_COORDS[[care]] ] = k -1
    k <- k + 1
}
k <- 1
for ( x in AGE_COMPS ){
    AGE[ AGE_COORDS[[x]] ] = k -1
    k <- k + 1
}
k <- 1
for ( x in NH_COMPS ){
    NH[ NH_COORDS[[x]] ] = k -1
    k <- k + 1
}
k <- 1
for ( x in RISK_COMPS ){
```

```
    RISK[ RISK_COORDS[[x]] ] = k -1
    k <- k + 1
}

## helpers
m <- length(DEMES)
# pr row transmission goes to col
prRecipMat <- matrix( 0. , nrow = m, ncol = m )
colnames(prRecipMat) = rownames(prRecipMat) <- DEMES
.mweight <- function( rowdeme, coldeme ){
    if (rowdeme=='src') return (0)
    if (coldeme=='src') return (0)
    rowage <-  as.numeric( regmatches( rowdeme, regexec( "\\.age([0-9])", rowdeme) )[[1]][2] )
    colage <-  as.numeric( regmatches( coldeme, regexec( "\\.age([0-9])", coldeme) )[[1]][2] )
    colpss <- as.numeric( regmatches( coldeme, regexec( "stage([0-9])", coldeme) )[[1]][2] )
    colcare <- as.numeric( regmatches( coldeme, regexec( "care([0-9])", coldeme) )[[1]][2] )
    colrisk <- as.numeric( regmatches( coldeme, regexec( "riskLevel([0-9])", coldeme) )[[1]][2] )
    wcare <- ifelse( colcare == 1, 1, 0)
    wrisk <- ifelse( colrisk == 1, theta['pRiskLevel1'], 1 - theta['pRiskLevel1'] )
#~  browser()
    if (colpss != 1) return(0)
    wrisk * wcare  * theta['age_assort_factor']^abs( rowage - colage )
}
for (i in 1:(m-1)) for (j in 1:(m-1)){
    prRecipMat[i,j] <- .mweight( DEMES[i], DEMES[j] )
}
prRecipMat <- prRecipMat / rowSums( prRecipMat )
prRecipMat[m,] <- 0
prRecipMat[m,m] <- 1

prStageRecipMat <- matrix( 0, nrow = m, ncol = m );
colnames(prStageRecipMat) = rownames(prStageRecipMat) <- DEMES
.stagemweight <- function(rowdeme, coldeme){
    if (rowdeme=='src') return (0)
    if (coldeme=='src') return (0)
    rowage <-  as.numeric( regmatches( rowdeme, regexec( "\\.age([0-9])", rowdeme) )[[1]][2] )
    colage <-  as.numeric( regmatches( coldeme, regexec( "\\.age([0-9])", coldeme) )[[1]][2] )
    rowstage <- as.numeric( regmatches( rowdeme, regexec( "stage([0-9])", coldeme) )[[1]][2] )
    colstage <- as.numeric( regmatches( coldeme, regexec( "stage([0-9])", coldeme) )[[1]][2] )
    rowcare <- as.numeric( regmatches( rowdeme, regexec( "care([0-9])", coldeme) )[[1]][2] )
    colcare <- as.numeric( regmatches( coldeme, regexec( "care([0-9])", coldeme) )[[1]][2] )
    rowrisk <- as.numeric( regmatches( rowdeme, regexec( "riskLevel([0-9])", coldeme) )[[1]][2] )
    colrisk <- as.numeric( regmatches( coldeme, regexec( "riskLevel([0-9])", coldeme) )[[1]][2] )
    if ( rowstage != 1 ) return(0)
    if (colage != rowage) return(0)
    if (colcare != rowcare ) return (0)
    if (colrisk!= rowrisk) return(0)
    return( pstarts[ colstage] )
}
for (i in 1:(m-1)) for (j in 1:(m-1)){
    prStageRecipMat[i,j] <- .stagemweight( DEMES[i], DEMES[j] )
}
prStageRecipMat <- prStageRecipMat/rowSums( prStageRecipMat )
```

```
prStageRecipMat[is.na(prStageRecipMat)] <- 0


## mig mat: deme indices of destination for transition in age, care and stage
# NOTE uses R indices
STAGEPROG_RECIP <- rep(-1, m)
CARE_RECIP <- rep(-1, m )
AGE_RECIP <- rep(-1, m )
#~ RISK_RECIP not needed
for (i in 1:(m-1)){
    deme <- DEMES[i]
    age <- as.numeric( regmatches( deme, regexec( "\\.age([0-9])", deme) )[[1]][2] )
    care <- as.numeric( regmatches( deme, regexec( "care([0-9])", deme) )[[1]][2] )
    stage <- as.numeric( regmatches( deme, regexec( "stage([0-9])", deme) )[[1]][2] )

    if (age < length(AGE_COMPS)){
        recip_age <- age + 1
        recip_age_deme <- sub( paste(sep='', '\\.age', age)
          , paste(sep='', '\\.age',  recip_age)
          , deme )
        AGE_RECIP[i] = which(DEMES==recip_age_deme)
    }

    if (care < length(CARE_COMPS) ){
        if (care==1 || stage > 2){ # NOTE cd4 threshold for treatment
            recip_care <- care + 1
            recip_care_deme <- sub( paste(sep='', 'care', care)
              , paste(sep='', 'care',  recip_care)
              , deme )
            CARE_RECIP[i] = which(DEMES==recip_care_deme)
        }
    }

    if (stage < length(NH_COMPS) ){
        recip_stage <- stage + 1
        recip_stage_deme <- sub( paste(sep='', 'stage', stage)
          , paste(sep='', 'stage',  recip_stage)
          , deme )
        STAGEPROG_RECIP[i] = which(DEMES==recip_stage_deme)
    }
}


## initial conditions
y0 <- setNames( rep(0, m ), DEMES )
y0[ CARE_COORDS$care1 ] <- 1 / length( CARE_COORDS$care1 )
y0[m] <- theta['src0'] # initial source size
```

- `prRecipMat` represents the probabilities that a recipient gets infection, conditioning on
  - risk level,
  - stage (only individuals in EHI stage 'get' infection),
  - care status (only undiagnosed 'get' infection) and
  - age assortativity (power of age class difference),

and intervenes in F matrix $F(i,j) = \text{incidence} \cdot w \cdot \text{prRecipMat}(i,j)$, with $w = \beta_{\text{stage}} \cdot \beta_{\text{age}} \cdot \beta_{\text{care}} \cdot \beta_{\text{risk}}$

- `prStageRecipMat` represents the probability for a recipient in EHI to jump to next other CD4 stage
- Natural mortality corresponds to aging out of age 4
- AIDS mortality corresponds to transitioning out of AIDS stage

```
#~ idea for hacking incidence and diagnosis rates(t)
#~ phillips incidence estimate -> scale so cuminf has about right value
#~ make diagnosis rate linear from zero; tune so that 80pc diagnosed in present
# incidence (t)
phil_inc <- data.matrix( read.table( 'incidence.tsv') )[,1]
phil_inc_times <- seq( 1980, 2010, length.out = length(phil_inc))
d_phil_inc_times <- phil_inc_times[2] - phil_inc_times[1]
phil_inc.t <- approxfun( phil_inc_times, phil_inc, rule = 2 )
inc.t <- function(t, theta) {
    y <- days2years(t)
    i <- min(length(phil_inc), max(1, 1 + floor( (y - phil_inc_times[1]) / d_phil_inc_times )) )
    phil_inc[i] * theta['inc_scale']
}


# diagnosis rates (t)
#~ phe_diags_total <- c( 23,    21,     71,     179,    646,    2938,   2648,   2385,   1940,   2169,
diag.t <- function(t, theta){
## NOTE return val needs to be rate in units of event per day
    y <- days2years( t)
    mdr <- theta['max_diag_rate'] #per year
    dr_accel <- theta['accel_diag_rate']
    dr85 <- theta[ 'diag_rate_85' ] /365 #per day
    if (y > 1985){
        return( max( dr85,  mdr / ( 1 + exp(-(y-1985) * dr_accel) ) / 365 )   ) #per day
    }
    dr85
}




# treatment rates (t)
tr.t <- function(t){
    y <- days2years(t)
    if ( y < 1995 ) return(0 )
     ( 1 / ( 1 + exp(-(y - 2e3)/2)) ) /365
}
#~ ys <- 1990:2012
#~ plot( ys, 1 / ( 1 + exp(-(ys - 2e3)/5)) )
```

Individual transmission rates are normalised so that total incidence is based on a previous study (Phillips et al. 2013) assuming that dynamics of new infections in MSM was the same at the country level and in London. Rates of treatment were modelled as zero before 1995 and then increase according to a logistic function with maximum 1 and steepness 0.5.

Initial parameter values for baseline scenario

Notation	Parameter	Value
	Age progression rate [a]	
α_1	Group 1 [18-27)	$1/9/365\ day^{-1}$
α_2	Group 2 [27-33)	$1/6/365\ day^{-1}$
α_3	Group 3 [33-40)	$1/7/365\ day^{-1}$
α_4	Group 4 [40-80.5)	$1/40.5/365\ day^{-1}$
	Stage progression rate[b]	
γ_1	Stage 1 (early HIV infection)	$1/0.5/365\ day^{-1}$
γ_2	Stage 2 (CD4 $>$ 500 cells/mm3)	$1/3.32/365\ day^{-1}$
γ_3	Stage 3 (350 $<$ CD4 $\leq$ 500 cells/mm3)	$1/2.7/365\ day^{-1}$
γ_4	Stage 4 (200 $<$ CD4 $\leq$ 350 cells/mm3)	$1/5.5/365\ day^{-1}$
γ_5	Stage 5 (CD4 $\leq$ 200 cells/mm3)	$1/5.06/365\ day^{-1}$
	Fraction of individuals transitioning from [b]	
π_1	Stage 1 to stage 2	0.76
π_2	Stage 1 to stage 3	0.19
π_3	Stage 1 to stage 4	0.05
π_4	Stage 1 to stage 5	0
a	Age assortativity factor	0.5
p	Proportion of individuals in low-risk group	0.8
m	Per lineage rate of migration to source compartment	$1/50/365\ day^{-1}$
g	Rate of growth of source compartment	$1/3/365\ day^{-1}$
s	Initial size of source compartment	1000
i	Incidence scaling factor for London MSM [c]	0.03
	Diagnosis rate	
d_{85}	Fixed rate prior to 1985	$1/10\ year^{-1}$
μ_d	Maximum value of logistic function after 1985 [c]	$1/3\ year^{-1}$
k_d	Steepness of logistic function after 1985 [c]	$1/7\ year^{-1}$
	Treatment rate	
t_{95}	Fixed rate prior to 1995	0
μ_t	Maximum value of logistic function after 1995	1
k_t	Steepness of logistic function after 1995	0.5
e	Treatment effectiveness	0.95
	Transmission weight conferred to individuals in	
w_{s1}	Stage 1	1
w_{s2} to w_{s4}	Stages 2 to 4	0.1
w_{s5}	Stage 5	0.3
w_{a1} to w_{a4}	Age groups 1 to 4	1
w_{c1}	Care status 1 (undiagnosed)	1
w_{c2}	Care status 2 (diagnosed and untreated)	0.5
w_{c3}	Care status 3 (diagnosed and treated)	0.05
w_{r1}	Risk status 1 (low risk)	1
w_{r2}	Risk status 2 (high risk)	10

[a] From quartiles of age of MSM diagnosed in London available in UKDRDB

[b] From Cori et al. AIDS 2015

[c] Initial value later calibrated to retrieve observed number of diagnosed cases from surveillance data

Source C functions

C code defines:

- 'Birth matrix' $F(t)$: Number of transmissions from donor in each of 120 compartments to recipient in each compartment over time
- 'Migration matrix' $G(t)$: Number of transition from each compartment to each compartment over time

`AGE_RECIP`, `CARE_RECIP` and `STAGEPROG_RECIP` are vectors of indices of compartments to which transition (migration) occurs

- With the form $G_{i,i+1}(t) = rateI_i$
- with migration from source compartment
- with reduced progression for treated individuals. When treated, speed of progression through stages is reduced by treatment effectiveness e: $\gamma_i(1-e)$

```
sourceCpp( 'model0.cpp' ) # F_matrix and G_matrix fns

dydt <- function(t,y, parms, ... ){
    y <- pmax(y, 0 )
    incidence <- inc.t( t, theta )
    care_rates <- c( diag.t( t, theta), tr.t( t) )
    FF <- F_matrix( incidence
      , y
      , as.list(theta)
      , DEMES
      , NH
      , AGE
      , CARE
      , RISK
      , nh_wtransm
      , age_wtransm
      , care_wtransm
      , risk_wtransm
      , prRecipMat
    )

    GG <- G_matrix( y
      , as.list(theta)
      , DEMES
      , NH
      , AGE
      , CARE
      , RISK
      , STAGEPROG_RECIP
      , AGE_RECIP
      , CARE_RECIP
      , stageprog_rates
      , age_rates
      , care_rates
      , prStageRecipMat
    )
    GGns <- GG
    GGns[m, ] = GG[, m ] <- 0

    dy <- colSums(FF) + colSums(GGns) - rowSums(GGns)
    names(dy) <- DEMES
    ## aids mort
    dy[NH_COORDS$stage5] <-  dy[NH_COORDS$stage5] -  y[ NH_COORDS$stage5 ] * stageprog_rates[5]
```

```
    ## nat mort
    dy[AGE_COORDS$age4] <-  dy[AGE_COORDS$age4] -  y[AGE_COORDS$age4] * age_rates[4]
    ## source
    dy[m] <- y[m] * theta['srcGrowthRate']

#~ if (days2years( t ) > 1995-.1 ) {
#~ print( days2years( t) )
#~ csf <- colSums( FF )
#~ csg <- colSums( GGns )
#~ rsg <- rowSums( GGns )
#~ sum( csf[ CARE_COORDS$care1 ] )
#~ sum( csg[ CARE_COORDS$care1 ] )
#~ sum( rsg[ CARE_COORDS$care1 ] )
#~ sum( GGns[ CARE_COORDS$care1, CARE_COORDS$care1 ] )
#~  browser()
#~ }

    list(dy )
}


.tfgy <- function(desolve){
# for input to tree simulator
    .t <- desolve[,1]
    .F <- lapply( 1:nrow(desolve), function(i){
        y <- desolve[i,-1]
        t <- desolve[i, 1]
        incidence <- inc.t( t, theta )
        FF <- F_matrix( incidence
          , y
          , as.list(theta)
          , DEMES
          , NH
          , AGE
          , CARE
          , RISK
          , nh_wtransm
          , age_wtransm
          , care_wtransm
          , risk_wtransm
          , prRecipMat
        )
        rownames(FF) = colnames(FF) <- DEMES
        FF
    })

    .G <- lapply( 1:nrow(desolve), function(i){
        y <- desolve[i,-1]
        t <- desolve[i, 1]
        care_rates <- c( diag.t( t, theta), tr.t( t) )
        GG <- G_matrix( y
          , as.list(theta)
          , DEMES
```

```
          , NH
          , AGE
          , CARE
          , RISK
          , STAGEPROG_RECIP
          , AGE_RECIP
          , CARE_RECIP
          , stageprog_rates
          , age_rates
          , care_rates
          , prStageRecipMat
        )
        rownames(GG) = colnames(GG) <- DEMES
        GG
    })

    .Y <- lapply( 1:nrow(desolve), function(i) {
        desolve[ i, -1 ]
    })

    list( .t, .F, .G, .Y )
}
```

Calibrate scale of incidence curve and diagnosis rate

Incidence and diagnosis rates are modeled as logistic functions of time and jointly calibrated to match the number of MSM living with diagnosed HIV in London in 2012 (Yin et al. 2014).

Notes: - incidence and diagnosis rate scaling factors are a priori. Now, there are fitted (doi:10.1371/journal.pone.0055312.g002 (Fig 2.A)) - iterations set to 30 for speed

```
if (T)
{
  #~ PHE: 15552 diagnosed msm in london in 2012
  propDiagnosed2012 <- 4/5
  I2012 <- 15552  / propDiagnosed2012 # assuming 80pc diagnosed
  #objfun based on both the above stats:
  objfun <- function( lntheta0 )
  {
    theta[ names(lntheta0) ] <<- exp(lntheta0) # using globals..
    o <- ode(y=y0, times=times_day, func=dydt, parms=list(), method = 'euler')
    ifin <- sum( o[nrow(o), 2:(ncol(o)-1) ] )
    idiagnosed <- sum( o[nrow(o), 1 + c( CARE_COORDS$care2, CARE_COORDS$care3) ] )
    print(paste( ifin, idiagnosed ))
    print( theta[fit_names ] )
    ((ifin - I2012) / I2012)^2 + (idiagnosed/ifin - propDiagnosed2012)^2
  }
  fit_names <- c('inc_scale', 'max_diag_rate', 'accel_diag_rate')
  theta_start <- log(theta[fit_names]) # default values
  o <- optim( theta_start, objfun , control = list(trace=6, maxit=30) ) # 300
  theta_docked <- exp(o$par)
  print((o))
}
```

```
##   Nelder-Mead direct search function minimizer
## [1] "8554.14435253098 6849.07424312644"
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0300000       0.3333333       0.1428571
## function value for initial parameters = 0.313569
##   Scaled convergence tolerance is 4.67254e-09
## Stepsize computed as 0.350656
## [1] "12146.8669121029 9725.66989684758"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.04259995      0.33333333      0.14285714
## [1] "8703.55304294158 7447.7129839425"
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0300000       0.4733328       0.1428571
## [1] "8582.64697185742 6906.74233092599"
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0300000       0.3333333       0.2028569
## BUILD              4 0.313569 0.140746
## [1] "10960.1212959755 9222.45113137943"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.03790064      0.42111818      0.18047922
## [1] "10305.6571750393 8575.86678393735"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.03574911      0.39721229      0.17023384
## LO-REDUCTION       6 0.311951 0.140746
## [1] "12848.3174427568 11022.8230534416"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.04429248      0.49213867      0.11757029
## [1] "15688.7796710922 13754.200063213"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.05381885      0.59798717      0.08950586
## EXTENSION          8 0.308124 0.043117
## [1] "18808.8387444176 15616.1284892184"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06539413      0.40502620      0.12224144
## [1] "27630.2960283457 22538.9219174215"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.09654894      0.37466349      0.11307763
## REFLECTION        10 0.191996 0.001969
## [1] "21381.8828908072 17736.615904828"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.07446320      0.44357306      0.07462552
## [1] "18100.5254360366 15118.1664094333"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06289525      0.43784951      0.09306199
## LO-REDUCTION      12 0.140746 0.001969
## [1] "25088.7358167917 22180.1512525301"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.08589836      0.67213188      0.07084777
## [1] "20955.5439780965 18188.2116054879"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.07208434      0.56404085      0.08442482
## LO-REDUCTION      14 0.043117 0.001969
## [1] "23579.3470220625 19051.786693464"
##       inc_scale   max_diag_rate accel_diag_rate
```

```
##      0.08261597      0.36034666      0.10875665
## [1] "17386.8580834944 14987.1864384658"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.05990551      0.52686461      0.09397288
## HI-REDUCTION      16 0.014996 0.001969
## [1] "21315.7004086346 17643.315081859"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.07422182      0.40899061      0.10358688
## [1] "20261.9438402299 16957.5690227224"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.07035021      0.43572227      0.10109489
## LO-REDUCTION      18 0.010694 0.001969
## [1] "17248.0074526735 13665.3551628611"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06068859      0.32163844      0.13001991
## [1] "19979.3630762675 17025.5360623877"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06904901      0.49014537      0.09404919
## HI-REDUCTION      20 0.005989 0.001969
## [1] "21383.6755224828 18058.6438284874"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.07402127      0.44670449      0.11878533
## [1] "18871.7385055006 15810.9410486219"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06550924      0.44004666      0.09891678
## HI-REDUCTION      22 0.003490 0.001969
## [1] "18638.4173250671 15209.5888431415"
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0650999       0.3713666       0.1215598
## [1] "17984.2966037943 14288.7020912861"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06321087      0.32325244      0.13819982
## REFLECTION        24 0.003151 0.001957
## [1] "17392.8408917522 14244.7750523858"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06067582      0.37553178      0.12787621
## [1] "19504.4051695296 16242.6658768222"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06779583      0.41982568      0.10721223
## HI-REDUCTION      26 0.002284 0.001085
## [1] "19084.6413749363 15526.6766141707"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06666719      0.36036110      0.13790031
## [1] "19180.0720508231 15320.7371811328"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06725382      0.32610502      0.16282192
## EXTENSION         28 0.001969 0.000180
## [1] "19407.4317963503 15574.8582245401"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06804477      0.33884657      0.13508683
## [1] "19704.8364472481 15513.0227516245"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06941011      0.30992962      0.14200718
## REFLECTION        30 0.001957 0.000009
```

```
## [1] "20125.6074286531 16269.4204285903"
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0703984       0.3476564       0.1457438
## [1] "19743.6517930224 16001.1927609259"
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06903466      0.35343811      0.13928044
## Exiting from Nelder Mead minimizer
##     32 function evaluations used
## $par
##       inc_scale   max_diag_rate accel_diag_rate
##       -2.687589       -1.082208       -2.001838
##
## $value
## [1] 9.158668e-06
##
## $counts
## function gradient
##       32       NA
##
## $convergence
## [1] 1
##
## $message
## NULL
```

Comparison of initial and calibrated values

```
print(exp(theta_start))
```

```
##       inc_scale   max_diag_rate accel_diag_rate
##       0.0300000       0.3333333       0.1428571
```

```
print(theta_docked)
```

```
##       inc_scale   max_diag_rate accel_diag_rate
##      0.06804477      0.33884657      0.13508683
```

```
##print(theta_default)
##print(theta)
```

Tree reconstruction

Genealogical trees are simulated conditioning on the epidemic history, and trees are matched to the real UK metadata data pertaining to times of sampling and clinical stage of infection.

TODO: `model0-simulateBaseline0.R` to run the simulation and generate a tree

References

Cori, Anne, Michael Pickles, Ard van Sighem, Luuk Gras, Daniela Bezemer, Peter Reiss, and Christophe Fraser. 2015. "CD4+ Cell Dynamics in Untreated HIV-1 Infection: Overall Rates, and Effects of Age, Viral Load, Sex and Calendar Time." *AIDS (London, England)* 29 (18): 2435–46. doi:10.1097/QAD.0000000000000854.

"HIV and STIs in Men Who Have Sex with Men in London." 2014. Public Health England. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/357451/2014_09_

17_STIs_HIV_in_MSM_in_London_v1_0.pdf.

Phillips, Andrew N., Valentina Cambiano, Fumiyo Nakagawa, Alison E. Brown, Fiona Lampe, Alison Rodger, Alec Miners, et al. 2013. "Increased HIV Incidence in Men Who Have Sex with Men Despite High Levels of ART-Induced Viral Suppression: Analysis of an Extensively Documented Epidemic." *PloS One* 8 (2): e55312. doi:10.1371/journal.pone.0055312.

Yin, Z, A E Brown, G Hughes, A Nardone, O N Gill, V C Delpech, and & contributors. 2014. "HIV in the United Kingdom 2014 Report: Data to End 2013." London: Public Health England. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/401662/2014_PHE_HIV_annual_report_draft_Final_07-01-2015.pdf.