Run simulated epidemic history

Stephane

3/27/2017

Read source code and execute chunk by chunk

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

Load packages

```
require(phydynR) # replaces rcolgem
require(deSolve)
require(Rcpp)
```

Define input parameters

```
##- 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 = 0.09401734 # based on docking (see below) # initial = .03
  , max_diag_rate = 0.66227809 # based on docking (see below) # initial = 1/3 (time 2 diag of 3 yrs)
  , diag_rate_85 = 1/10
  , accel_diag_rate = 0.03196171 # based on docking (see below ) # initial = 1/7 # accel of logistic fu
  , treatmentEffectiveness = .95 # slows stage progression
  , pstarts
    , age_rates
    , stageprog_rates
)

theta_default <- theta
```

```r
##- 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
```

Initial parameter values for baseline scenario

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

[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 ca surveillance data

Note: incidence and diagnosis rate scaling factors are a priori. Now, there are fitted doi:10.1371/journal.pone. 0055312.g002 (Fig 2.A)

every individuals start infection at EHI stage

prRecipMat: - prob that recipient get infection, conditionning on - prob of being risk level 1 (80%) vs risk level 2 (20%) - EHI stage (only those recipient get infection) - care status (only undiagnosed get infection) - age

assortativity (power of age class difference) - intervenes in F matrix [F(i,j) = incidence * w * prRecipMat(i,j), with w = beta_NH * beta_age * beta_care * beta_risk]

prStageRecipMat: - Prob for EHI recipient to jump to next other CD4 stage

Source C functions

'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

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


## solve model
#~ F_matrix( double incidence
#~   , NumericVector sizes
#~   , NumericVector theta
#~   , CharacterVector demes
#~   , IntegerVector NH // length m indicators for each deme
#~   , IntegerVector AGE
#~   , IntegerVector CARE
#~   , IntegerVector RISK
#~   , NumericVector nh_wtransm // associated weight for each category
#~   , NumericVector age_wtransm
#~   , NumericVector care_wtransm
#~   , NumericVector risk_wtransm
#~   , NumericMatrix prRecipMat // pstartstage & age mixing & prisklevel
#~ )
#~ G_matrix( NumericVector sizes
#~   , NumericVector theta
#~   , CharacterVector demes
#~   , IntegerVector NH // length m indicators for each deme
#~   , IntegerVector AGE
#~   , IntegerVector CARE
#~   , IntegerVector RISK
#~   , IntegerVector stageprog_recip // destination for migration
#~   , IntegerVector age_recip
#~   , IntegerVector care_recip
#~   , NumericVector stageprog_rates //rates for each deme
#~   , NumericVector age_rates
#~   , NumericVector care_rates // note these depend on time
#~ )

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 )
}
```