

```

##### R Code for 'Other key life history traits from MPMs' #####

# https://compadre-db.org/Education/article/other-key-life-history-traits-from-mpms

# Compiled on 9 February 2021

##### Preliminaries #####

# Use the command following commands if you haven't already
# downloaded the packages:

# install.packages("popbio")
# install.packages("MASS")

library(popbio)
library(MASS)

##### Other key life history traits from MPMs #####

# Define function
lifeTimeRepEvents <- function(matU, matF, startLife = 1) {

  uDim = dim(matU)[1]
  surv = colSums(matU)
  repLifeStages = colSums(matF)
  repLifeStages[which(repLifeStages>0)] <- 1

  if(missing(matF) | missing(matU)) {stop('matU or matF missing')}
  if(sum(matF, na.rm=T)==0) {stop('matF contains only 0 values')}

  # Probability of survival to first reprod event
  Uprime <- matU
  Uprime[, which(repLifeStages==1)] <- 0
  Mprime = matrix(0,2,uDim)
  for (p in 1:uDim[1]) {
    if (repLifeStages[p]==1) {Mprime[2,p] = 1} else {
      Mprime[1,p] = 1-surv[p]
    }
  }
  Bprime = Mprime%*(ginv(diag(uDim)-Uprime))
  pRep = Bprime[2,startLife]

  out = data.frame(pRep = pRep)

  # Age at first reproduction (La; Caswell 2001, p 124)
  D = diag(c(Bprime[2,]))
  Uprimecond = D%*Uprime%*ginv(D)
  expTimeReprod = colSums(ginv(diag(uDim)-Uprimecond))

```

```

La = expTimeReprod[startLife]

out$La = La

# Mean life expectancy conditional on
# entering the life cycle in the first reproductive stage
firstRepLifeStage = min(which(repLifeStages==1))
N = solve(diag(uDim[1])-matU)
meanRepLifeExpectancy = colSums(N)[firstRepLifeStage]

out$meanRepLifeExpectancy = meanRepLifeExpectancy

# Life expectancy from mean maturity
remainingMatureLifeExpectancy = colSums(N)[startLife]-La

out$remainingMatureLifeExpectancy = remainingMatureLifeExpectancy

return(out)
}

# Create example matrices
matU <- matrix(c(0.1,0.1,0.3,0.1,0.3,0.3,0.1,0.2,0.3,0.3,0.2,0.2,0,0.2,0.3,0.2),
              nrow = 4, ncol = 4, byrow = T)

colSums(matU) # No column can be > 1

matF <- matrix(c(0,0.3,40,30, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
              nrow = 4, ncol = 4, byrow = T)

matA <- matU + matF

# Calculate life history traits
dyn <- eigen.analysis(matA)
dyn$lambda1

dyn$stable.stage

dyn$repro.value

N <- solve(diag(dim(matU)[1])-matU)
N

colSums(N)

# Calculate net reproductive rate and generation time
R <- matF %*% N

Ro <- lambda(R)

```

```
lambda <- lambda(matA)
```

```
generation.time <- log(Ro)/log(lambda)
```

```
Ro
```

```
generation.time
```

```
# Calculate life history traits using defined function  
lifeTimeRepEvents(matU, matF, startLife = 2)
```