List of Example Scripts

Script 1. BlowFlies.R (Section 3.1): Single Species with fixed death rates and stage durations.

```
library(stagePop)
#All the vectors are specified in the order of the life cycle
#e.g. start with eggs and finish with reproducing adults

solver.options=list(DDEsolver='deSolve', atol=1e-3, rtol=1e-3,
    hbsize=1e4)
#solver.options=list(DDEsolver='PBS', tol=1e-7, hbsize=1e4, dt=0.01)

blowFliesFunctions <- list(
                        reproFunc=function(x,time,species,strain){
                          A0=600
                          q=8.5
                          reprod=q*x$blowflies['adults',1] *
                              exp(-x$blowflies['adults',1]/A0)
                          return(reprod)
                        },
                        deathFunc=function(stage,x,time,species,strain){
                        #per capita death rate (/d)
                          a=c(0.07,0.004,0.003,0.0025,0.27)
                          return(a[stage])
                        },
                        durationFunc=function(stage,x,time,species,strain){
                        #duration of each stage in days
                          a=c(0.6,5.0,5.9,4.1)
                          return(a[stage])
                        },
                        immigrationFunc=function(stage,x,time,species,strain){
                          v=0
                          if (stage==5){
                            if (time>=0 & time<=1){v=100}
                          }
                          return(v)
                        },
                        emigrationFunc=function(stage,x,time,species,strain){return(0)}
                        )

modelOutput = popModel(
  numSpecies=1,
  numStages=5,
  ICs=list(matrix(0,nrow=5,ncol=1)),
  timeVec=seq(0,200,0.5),
  timeDependLoss=TRUE,
  timeDependDuration=FALSE,
  rateFunctions=blowFliesFunctions,
  solverOptions=solver.options,
  stageNames=list(c('eggs','larvae','pupae','juveniles','adults')),
  speciesNames=c('blowflies'),
  saveFig=TRUE,
  figType='eps',
  figName='blowflies'
  )
```

Script 2. LarvalComp.R (Section 3.2): A single species with density-dependent death rates.

```
library(stagePop)

solver.options=list(DDEsolver='deSolve',atol=1e-4,rtol=1e-4,method='lsoda',hbsize=1e6)
#solver.options=list(DDEsolver='PBS',tol=1e-6,hbsize=1e3,dt=0.1)

case=1 #choose case (1 or 2)
if (case==1) {
  num.stages=2
  stage.names=c('larvae','adults')
} else{
  num.stages=3
  stage.names=c('larvae','adults','dead adults')
}

larvalCompFunctions <- list(
                          reproFunc=function(x,time,species,strain){
                            reprod=9.4*x$flies['adults',1]
                            return(reprod)
                          },
                          deathFunc=function(stage,x,time,species,strain){
                            if (stage==1){v=5e-5*x$flies['larvae',1]}
                            if (stage>=2){if (case==1){v=0.2}else{v=0}}
                            return(v)
                          },
                          durationFunc=function(stage,x,time,species,strain){
                            a=c(28,5)
                            return(a[stage])
                          },
                          immigrationFunc=function(stage,x,time,species,strain){
                            v=0
                            if (time>=0 & time<=1){
                              if (stage==2){v=20}}
                            return(v)
                          },
                          emigrationFunc=function(stage,x,time,species,strain){return(0)}
                          )

modelOutput=popModel(
  numSpecies=1,
  numStages=num.stages,
  ICs=list(matrix(0,nrow=num.stages,ncol=1)),
  timeVec=seq(0,500,0.5),
  timeDependLoss=TRUE,
  solverOptions=solver.options,
  rateFunctions=larvalCompFunctions,
  stageNames=list(stage.names),
  speciesNames='flies',
  saveFig=TRUE,
  figType='eps',
  figName=paste('LarvalComp',case,sep=''))
```

Script 3. VarDurEnv.R (Section 3.3): A single species whose stage durations depend on temperature

```
solver.options=list(DDEsolver='deSolve',atol=1e-6,rtol=1e-6,hbsize=1e5)
#solver.options=list(DDEsolver='PBS',tol=1e-8,hbsize=1e4,dt=0.01)

tempFunc=function(time){
  T=15*(1-cos(2*pi*(time+80)/365))
  return(T)}

tauFunc=function(T){
  maxDur=200; minDur=60
  v=min(minDur+((T-20)/2)^2,maxDur)
  return(v)}

varDurEnvFunctions<-list(
                       reproFunc=function(x,time,species,strain){
                         A0=600; q=11.5
                         reprod=q*x$Nematodes['adults',1] *
                             exp(-x$Nematodes['adults',1]/A0)
                         return(max(0,reprod))
                       },
                       deathFunc=function(stage,x,time,species,strain){
                         a=c(0.05,0.05)
                         v=a[stage]
                         return(max(0,v))
                       },
                       develFunc=function(stage,x,time,species,strain){
                         T=tempFunc(time)
                         v=1/tauFunc(T)
                         return(v)
                       },
                       durationFunc=function(stage,x,time,species,strain){
                         if (time==0){
                           T=tempFunc(time)
                           v=tauFunc(T)}
                         return(v)
                       },
                       immigrationFunc=function(stage,x,time,species,strain){
                         v=0
                         if (stage==2){if (time>=0 & time <=0.1){v=1}}
                         return(v)
                       },
                       emigrationFunc=function(stage,x,time,species,strain){return(0)}
)

modelOutput=popModel(
  numSpecies=1,numStages=2,
  timeDependLoss=FALSE,timeDependDuration=TRUE,
  ICs=list(matrix(0,nrow=2,ncol=1)),
  timeVec=seq(0,365*6,1),
  solverOptions=solver.options,
  rateFunctions=varDurEnvFunctions,
  stageNames=list(c('juveniles','adults')),speciesNames=c('Nematodes'))
```

Script 4. PredPrey1.R (Section 3.4): Classic Predator-Prey model.

```
growthRatePred=10
growthRatePrey=1
deathRatePrey=1
deathRatePred=0.5

#solver.options=list(DDEsolver='deSolve',atol=1e-6,rtol=1e-6,method='lsoda',hbsize=1e4)
solver.options=list(DDEsolver='PBS',tol=1e-7,hbsize=1e4,dt=0.01)

ppFunctions <- list(
                  reproFunc=function(x,time,species,strain){
                    if(species==1){
                      reprod=growthRatePrey*x$prey['adults',1]}
                    if(species==2){
                      reprod=growthRatePred*x$prey['adults',1]*x$predator['adults',1]}
                    return(max(0,reprod))
                  },
                  deathFunc=function(stage,x,time,species,strain){
                    if(species==1){
                      v=deathRatePrey*x$predator['adults',1]}
                    if(species==2){
                      v=deathRatePred}
                    return(max(0,v))
                  },
                  durationFunc=function(stage,x,time,species,strain){return(0)},
                  immigrationFunc=function(stage,x,time,species,strain){return(0)},
                  emigrationFunc=function(stage,x,time,species,strain){return(0)}
                  )

modelOutput=popModel(
  numSpecies=2,
  numStages=c(1,1),
  timeDependLoss=c(TRUE,FALSE),
  timeDependDuration=c(FALSE,FALSE),
  ICs=list(matrix(0.3,1,1),matrix(1,1,1)),
  timeVec=seq(0,100,0.1),
  solverOptions=solver.options,
  plotFigs=TRUE,
  rateFunctions=ppFunctions,
  speciesNames=c('prey','predator'),
  stageNames=list('adults','adults')
)
```

Script 5. PredPrey2.R (Section 3.4): Predator-Prey model with stage-structured predator

```
growthRatePred=10; growthRatePrey=1
deathRatePrey=1; deathRatePred=c(1.0,0.5)
preyCarryCapacity=1

#solver.options=list(DDEsolver='deSolve',atol=1e-6,rtol=1e-6,method='lsoda',hbsize=1e4)
solver.options=list(DDEsolver='PBS',tol=1e-8,hbsize=1e4,dt=0.01)

case=1 #choose case
if (case==1){juvPredDuration=0.1;lenTime=100}
if (case==2){juvPredDuration=0.1;lenTime=300}
if (case==3){juvPredDuration=1.8;lenTime=300}
if (case==4){juvPredDuration=0.1;lenTime=400;deathRatePred[1]=0}
if (case==5){juvPredDuration=5;lenTime=400;deathRatePred[1]=0}
if (case==6){juvPredDuration=15;lenTime=400;deathRatePred[1]=0}
if (case==7){juvPredDuration=20;lenTime=400;deathRatePred[1]=0}

ppFunctions <- list(
                  reproFunc=function(x,time,species,strain){
                    if(species==1){reprod=growthRatePrey*x$Prey['adult',1]}
                    if(species==2){reprod=growthRatePred*x$Prey['adult',1]*x$Predator['adult',1]}
                    return(max(0,reprod))
                  },
                  deathFunc=function(stage,x,time,species,strain){
                    if(species==1){
                      if(case==1){v=deathRatePrey*x$Predator['adult',1]}
                      if(case>1){v=deathRatePrey*x$Predator['adult',1]+
                        growthRatePrey*x$Prey['adult',1]/preyCarryCapacity}}
                    if(species==2){v=deathRatePred[stage]}
                    return(max(0,v))
                  },
                  durationFunc=function(stage,x,time,species,strain){
                    return(juvPredDuration)
                  },
                  immigrationFunc=function(stage,x,time,species,strain){return(0)},
                  emigrationFunc=function(stage,x,time,species,strain){return(0)}
                  )
modelOutput=popModel(
  numSpecies=2,
  numStages=c(1,2),
  timeDependLoss=c(TRUE,FALSE),
  timeDependDuration=c(FALSE,FALSE),
  ICs=list(matrix(0.3),matrix(c(0,1),nrow=2,ncol=1)),
  timeVec=seq(0,lenTime,0.1),
  solverOptions=solver.options,
  plotFigs=TRUE,
  rateFunctions=ppFunctions,
  stageNames=list('adult',c('juvenile','adult')),
  speciesNames=c('Prey','Predator')
  )
```

Script 6. Briggs.R (Section 3.5): Host-Parasitoid model (1 host, 2 parasitoids)

```
attackRateP=1; attackRateQ=2;
TE=0.5; TL=0.5; TJP=0.4; TJQ=0.4
deathE=0.1;deathL=0.1;deathA=0.1;deathJP=0.1;
deathP=8.0;deathJQ=0.1;deathQ=8.0
rho=33 #total lifetime fecundity
LstarQ=4.16;AstarQ=9.44;Qstar=3.40

BriggsFunctions <- list(
             reproFunc=function(x,time,species,strain){
               if (species==1){reprod=rho*deathA*x$Host['adults',1]}
               if (species==2){reprod=attackRateP*x$'Egg
                    Parasitoid'['adults',1]*x$Host['eggs',1]}
               if (species==3){reprod=attackRateQ*x$'Larval
                    Parasitoid'['adults',1]*x$Host['larvae',1]}
               return(max(0,reprod))
             },
             deathFunc=function(stage,x,time,species,strain){
               if (species==1){a=c(deathE,deathL,deathA);v=a[stage]
                   if (stage==1){v=a[stage]+attackRateP*max(x$'Egg
                        Parasitoid'['adults',1],0)}
                   if (stage==2){v=a[stage]+attackRateQ*max(x$'Larval
                        Parasitoid'['adults',1],0)}}
               if (species==2){a=c(deathJP,deathP);v=a[stage]}
               if (species==3){a=c(deathJQ,deathQ);v=a[stage]}
               return(max(0,v))
               },
              durationFunc=function(stage,x,time,species,strain){
               if (species==1){a=c(TE,TL)}
               if (species==2){a=TJP}
               if (species==3){a=TJQ}
               return(a[stage])
              },
              immigrationFunc=function(stage,x,time,species,strain){
               v=0
               if (species==1){if (time>=0 &
                    time<=0.1){f1=rho*deathA*AstarQ
                 if (stage==1){v=f1}
                 if (stage==2){v=f1*exp(-deathE*TE)}
                 if (stage==3){v=f1*exp(-deathE*TE-deathL*TL)}}}
               if(species==2){if(time>=20 & time<=20.1){
                   if(stage==2){v=1}}}
               if(species==3){if(time>=0 & time<=0.1){
                   if(stage==1){v=attackRateQ*Qstar*LstarQ}}}
               return(v)
              },
              emigrationFunc=function(stage,x,time,species,strain){return(0)}
              )

modelOutput=popModel(numSpecies=3,
  numStages=c(3,2,2),
  timeVec=seq(0,50,0.1),
```

```
  rateFunctions=BriggsFunctions,
  timeDependLoss=c(TRUE,FALSE,FALSE),
  timeDependDuration=c(FALSE,FALSE,FALSE),
  ICs=list(matrix(0,nrow=3,ncol=1),matrix(0,nrow=2,ncol=1),matrix(0,nrow=2,ncol=1)),
  solverOptions=list(DDEsolver='PBS',tol=1e-7,hbsize=1e4,dt=0.01),
  speciesNames=c('Host','Egg Parasitoid','Larval Parasitoid'),
  stageNames=list(c('eggs','larvae','adults'),c('eggs','adults'),c('eggs','adults'))
)
```

Script 7. VarDurFood.R (Section 3.6): Consumer-resource problem with variable stage duration.

```
fs=1; fmax=3; F0=0.1 #rate of food supply; max rate; initial food density
m=1 #number of mass units a larva must increase to become an adult
epsilon=1#const of proportionality between development and food
    consumption
K=1 #half sat constant for food consumption
q=5; dA=2 #reproduction rate; adult death rate
dL=log(q/dA) #larval death rate

solver.options=list(DDEsolver='deSolve',atol=1e-9,rtol=1e-9,method='lsoda',hbsize=1e4)
#solver.options=list(DDEsolver='PBS',tol=1e-9,hbsize=1e4,dt=0.01)

varDurFoodFunctions <- list(
                        reproFunc=function(x,time,species,strain){
                          if(species==1){reprod=fs}
                          if(species==2){reprod=q*x$Damselfly['adults',1]}
                          return(max(0,reprod))
                        },
                        deathFunc=function(stage,x,time,species,strain){
                          if(species==1){v=fmax*x$Damselfly['larvae',1]/(K+x$Food[1,1])}
                          if(species==2){a=c(dL,dA);v=a[stage]}
                          return(max(0,v))
                        },
                        durationFunc=function(stage,x,time,species,strain){
                          if(time==0 & species==2 & stage==1){
                            v=m/(epsilon*fmax*F0/(K+F0))}
                          return(v)
                        },
                        develFunc=function(stage,x,time,species,strain){
                          if (species==2 & stage==1){
                            v=epsilon*fmax*x$Food[1,1]/(K+x$Food[1,1])}
                          return(v)
                        },
                        immigrationFunc=function(stage,x,time,species,strain){
                          v=0
                          if (species==2 & stage==1){
                            if (time>=0 & time<=0.1){v=1}}
                          return(v)
                        },
                        emigrationFunc=function(stage,x,time,species,strain){return(0)}
                        )

modelOutput=popModel(
  numSpecies=2,speciesNames=c('Food','Damselfly'),
  numStages=c(1,2),stageNames=list('one',c('larvae','adults')),
  numStrains=c(1,1),
  timeDependLoss=c(TRUE,FALSE),timeDependDuration=c(FALSE,TRUE),
  ICs=list(matrix(F0,1,1),matrix(0,nrow=2,ncol=1)),
  timeVec=seq(0,30,0.1),
  solverOptions=solver.options, rateFunctions=varDurFoodFunctions
)
```

Script 8. MultipleStrains.R (Section 3.7): Consumer-resource with multiple strains which have a trade-off between maximum growth rate and stage duration.

```
Rin=10; V=1; K=1;Yield=0.5; num.strains=6
if (num.strains>1){Gmax=2+seq(1,num.strains)}else{Gmax=2}

case=2#choose case
if(case==1){num.stages=1;stage.names='reproductive';start=0.1}
if(case==2){num.stages=2;stage.names=c('lagged','reproductive');start=c(0,0.1)}

strainsFunctions <- list(
            reproFunc=function(x,time,species,strain){
              if (species==1){reprod=Rin*V}
              if (species==2){reprod=x$Bacteria['reproductive',strain]*
                  Gmax[strain]*x$Resource['food',1]/(x$Resource['food',1]+K)}
              return(reprod)
            },
            deathFunc=function(stage,x,time,species,strain){
              if (species==1){uptake=0*seq(1,num.strains)
                for (s in seq(1,num.strains)){
                  uptake[s]=(Gmax[s]/(x$Resource['food',1]+K))*(x$Bacteria['reproductive',s]/Yield)}
                death=sum(uptake)+V}
              if (species==2){
                if (stage==1){if(num.stages==2){death=0}else{death=V}}
                if (stage==2){death=V}}
              return(death)
            },
            durationFunc=function(stage,x,time,species,strain){
              durations=2*seq(1,num.strains)
              return(durations[strain])
            },
            immigrationFunc=function(stage,x,time,species,strain){return(0)},
            emigrationFunc=function(stage,x,time,species,strain){return(0)}
            )

modelOutput = popModel(
  numSpecies=2,
  numStrains=c(1,num.strains),
  numStages=c(1,num.stages),
  ICs=list(matrix(Rin,nrow=1,ncol=1),matrix(start,nrow=num.stages,ncol=num.strains)),
  timeVec=seq(0,100,0.5),
  timeDependLoss=c(TRUE,FALSE),
  timeDependDuration=c(FALSE,FALSE),
  rateFunctions=strainsFunctions,
  solverOptions=list(DDEsolver='PBS',tol=1e-7,hbsize=1e4,dt=0.01),
  stageNames=list(c('food'),stage.names),
  speciesNames=c('Resource','Bacteria'),
  saveFig=TRUE,figType='eps',figName=paste('multiStrain',case,sep=''),
  sumOverStrains=FALSE,
  plotStrainsFig=TRUE,saveStrainsFig=TRUE,strainsFigType='eps',strainsFigName='strainFig'
  )
```