Examples using spatibm

Peter Whigham

2019-08-30

Introduction

This vignette covers some example approaches to using the spatibm package for exploring questions regarding allele fixation, population survival and size.

Modelling Fixation of a neutral allele

Modelling of neutral alleles in a population allows a measure of inbreeding and loss of diversity. For the *spatibm* package this is influenced by the breeding, survival, fecundity and the structure of the habitat and spatial density of individuals. A function has been provided that,given an initial population, runs a population until diversity is lost within the population, or a set number of generations has been reached. Multiple runs of this function allow a confidence interval for fixation to be produced. The default definitions for movement, etc. provided with *create.ibm.population* will be used to simplify this description. A single rare allele with initial proportion 0.05 will be used to ensure fixation is reasonably quick.

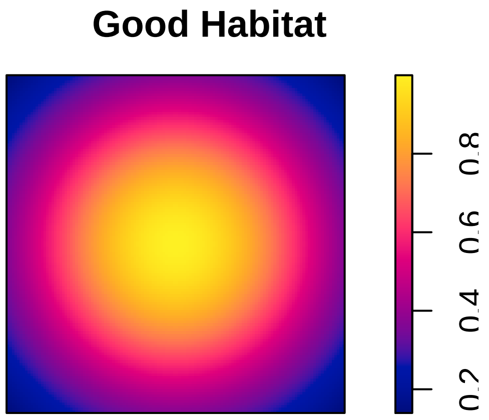
```
obs.win <- owin(xrange=c(-30,30),yrange=c(-30,30),unitname="metres")
allele.prop <- c(0.05)  # Make one rare allele (one locus, 2 alleles)
pop <- create.ibm.population(N=20,
                             spat.layout="random",
                             obs.win,
                             prob.females=0.5,
                             age.distribution=c(10,3),
                             allele.prop=allele.prop)

# Now expand the space
#
pop$window <- owin(xrange=c(-100,100), yrange=c(-100,100),unitname="metres")
hab.surface <- as.im(circle.habitat,pop$window,b=0.0001)
hab.list <- list(list(hab.surface,Inf,0)) # Just the one surface
breed.table <- breeding.table() # default values
survive.table <- survival.table() # default values
move.table <- movement.table() # default values
max.dist <- 10.0 # Breeding occurs for parents within 10 mtrs


fixed <- fixation(max.gens=30,
                  pop,
                  move.table,
                  survive.table,
                  breed.table,
                  habitat.list=hab.list, # stop overpopulation
                  crowd.table=NA, # No density dependence for survival
                  crowding.sigma=0.0,
                  max.dist=max.dist,
                  trace.output=TRUE)
#> Gen: 0 N=20 A:39..Gen:1 N=25 A:46..Gen:2 N=33 A:61..Gen:3 N=33 A:60..
#> Gen:4 N=28 A:48..Gen:5 N=23 A:41..Gen:6 N=22 A:39..
#> Gen:7 N=21 A:38..Gen:8 N=14 A:24..Gen:9 N=11 A:16..
#> Gen:10 N=16 A:21..Gen:11 N=21 A:26..Gen:12 N=16 A:18..
#> Gen:13 N=19 A:20..Gen:14 N=13 A:13..Gen:15 N=10 A:10..
#> Gen:16 N=7 A:7..Gen:17 N=8 A:8..Gen:18 N=6 A:6..
#> Gen:19 N=4 A:4..Gen:20 N=2 A:2..Gen:21 N=2 A:2..
#> Gen:22 N=2 A:2..Gen:23 N=2 A:2..Gen:24 N=1 A:1..
#> Gen:25 N=0 A:0..

fixed
#> [1]  1 25  0  0  0  0
```

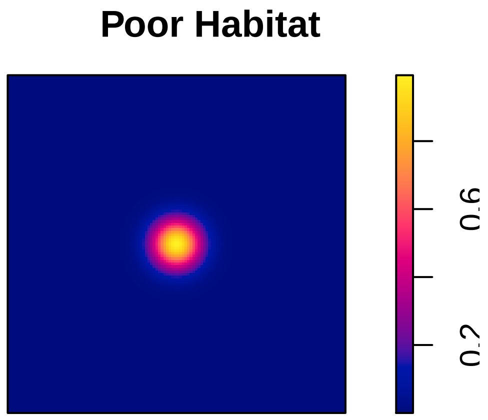
The result of a single call to fixation is a six element vector, showing True/False for fixation, the generation it occurred, the population count at the time of fixation, the sum of alleles, the number of males in the population, and the number of females. This is required to distinguish fixation due to the loss of the population (population count = 0), halting because the maximum number of generations has been reached, and halting because there are no viable breeding pairs (i.e. loss of diversity of males/females). Note that the sum of alleles is a measure of the remaining diversity if the population has not fixed. For a fixed population this will either be zero (all loci fixed at zero), or [num of loci*popsize] when all loci are fixed at one. Probability of fixation requires this model to be run repeatedly, and with max.gens set typically quite high; normally for a range of population sizes. This is best done as a script run (probably) overnight, and saving the results to a file each time the model is run.

Modelling a bottleneck

A bottleneck implies a restriction in the stability of the population. One appraoch to create this behaviour is to use (at least) 2 habitats - one which is reasonably good, and one that has a low probability of survival. The timing of these will determine the behaviour of the population and survival pattern. For example, let's create two circular habitats:

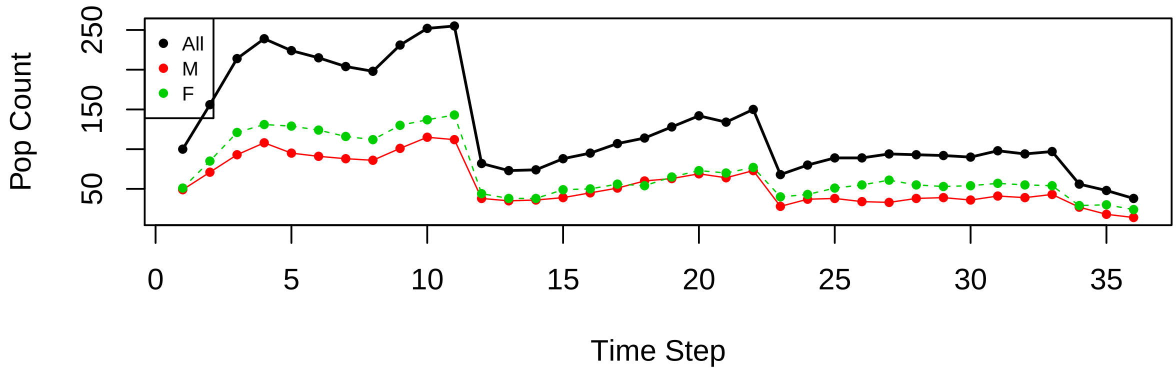
```
hab.win <- owin(xrange=c(-100,100),yrange=c(-100,100),unitname="metres")
good.hab <- as.im(circle.habitat,hab.win,b=0.0001) # Higher probability
poor.hab <- as.im(circle.habitat,hab.win,b=0.005) # Lower probability
par(mar=c(1,1,2,0))
plot(good.hab,main="Good Habitat")
plot(poor.hab,main="Poor Habitat")
```





The model can now be run with the poor habitat being introduced for one generation every 10 generations. Note of course that this can be changed by creating a more complex description of the habitat ordering.

```
pop <- create.ibm.population(N=100,
                              spat.layout="random",
                              obs.win,
                              prob.females=0.5,
                              age.distribution=c(20,3),
                              allele.prop=allele.prop)
pop$window <- hab.win
bottleneck <- multiple.step.habitat(steps=35,
                  pop, # Use population from previous example
                  move.table,
                  survive.table,
                  breed.table,
                  habitat.list=list(list(good.hab,10,0),list(poor.hab,1,0)),
                  crowd.table=NA, # No density dependence for survival
                  crowding.sigma=0.0,
                  max.dist=max.dist,
                  track.ids=FALSE,
                  trace.output=FALSE)

plot.pop.count(bottleneck)
```



Determining the population size for survival

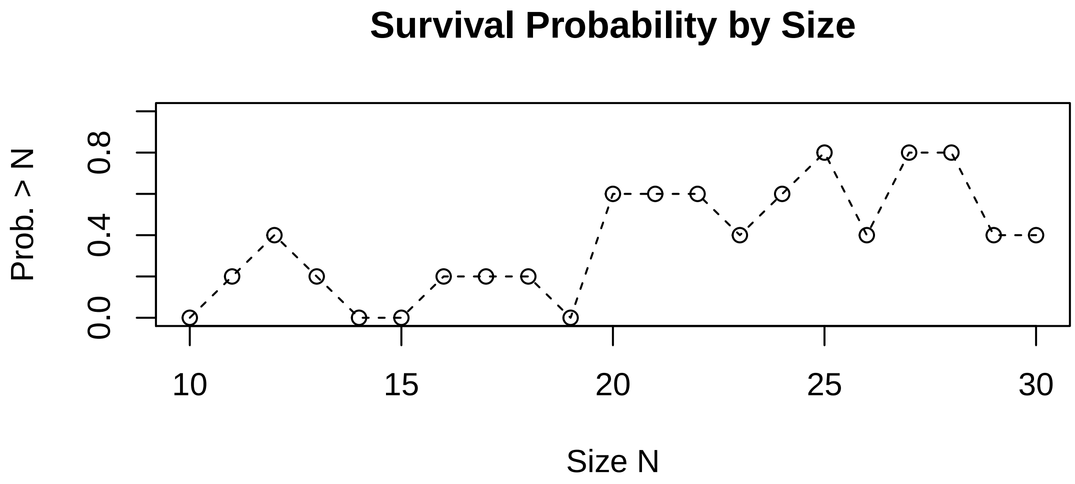
An important question to ask is what is a viable intial population size? One way of forming this question is to assume that a population commences with a size N and ask: What is the probability that the population size will be at least as large after G generations? Here we would assume that if a population size is less than N after a reasonable length of time that the long-term probability of survival is low.

```
# Show use of competitive release habitat
# circular habitat for competitive release stage
CRhabitat.surface <- as.im(circle.habitat,hab.win,b=0.0001)
# circular habitat 'normal level'
habitat.surface <- as.im(circle.habitat,hab.win,b=0.0002)
```

```
hab.list <- list(list(CRhabitat.surface,25,0),
                 list(habitat.surface,Inf,0))

surv <- popsize.survival(pop.sizes=10:30,steps=40,samples=5,
                         spat.layout="random",
                         obs.win,
                         prob.females=0.5, # Initial pop parameters
                         age.distribution=c(5,3),
                         allele.prop=c(0.5),
                         habitat.win=hab.win,
                         move.table,
                         survive.table,  # Behaviour parameters
                         breed.table,
                         habitat.list=hab.list, # Multiple habitats
                         crowd.table=NA,
                         crowding.sigma=0.0,
                         max.dist=10.0,
                         trace.output=FALSE
                         )

knitr::kable(head(surv,6))
```

N	finalN
10	0
10	0
10	0
10	0
10	0
11	0

```
# Now interpret finalN > N ?  as probability of viability
# Just use the number of times the population is greater than the
# initial population size after <steps> generations as the measure of
# viability.

N.vals <- unique(surv[,1])
probs <- NULL
for (N in N.vals)
{
  runs <- surv[which(surv[,1]==N),]
  probs <- c(probs,length(which(runs[,2]>runs[,1]))/nrow(runs))
}
plot(x=N.vals,y=probs,main="Survival Probability by Size",ylim=c(0,1),
     xlab="Size N",ylab="Prob. > N")
lines(x=N.vals,y=probs,lty=2)
```



The plot above shows a single measure of probability - of course it would be possible to bootstrap this value for a confidence interval and also to use more samples to produce a better estimate (if you are willing to wait).