

Getting started with spatibm

Peter Whigham

2019-08-30

Introduction

The spatibm package is designed to build a model of organism movement and interaction through a defined spatial region. The organisms are assumed to be dioecious, diploid with defined male and female roles. The behaviour of both sexes are independently defined. Although many packages exist for modelling movement in space and the related genetic diversity, most define a fixed carrying capacity (i.e. a maximum population size). The spatibm package offers a more appropriate mechanism for carrying capacity based on survival properties due to age, sex, habitat and crowding (resource constraints). Although this means the model is more complex to define, the resulting behaviour appears more realistic, emphasising the stochastic nature of populations. A simple diploid model of genetics is incorporated to allow questions of fixation for neutral alleles to be considered. The package has a range of functions that allow the user to construct complex models and answer questions that are not constrained by the original models, although there are limitations that will be discussed later. The package uses the R package spatstat for handling point data.

Creating a population

Individuals are defined as a set of points (i.e. they have no area) and have the following properties:

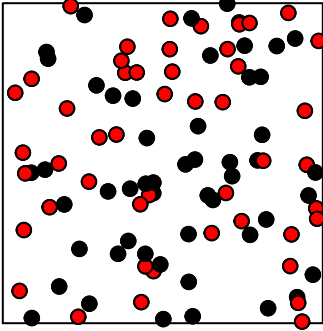
- Sex (male or female)
- Age
- Movement behaviour (based on age class)
- Survival properties (based on age class)
 - Survival probability as a baseline survival measure
 - Survival due to crowding
 - Survival modified by spatial habitat
- Breeding behaviour (based on age class)

The initial population is created with a population size N , a spatial layout, an observation window (the initial release window), the probability of being female, the age distribution for males/females and the allele frequency (probability) for each locus. The initial window is defined as an owin from the spatstat package. For example, a population of 100 individuals, with approximately a 50% split of males/females, initial age distribution $N(\mu=10, \sigma=3)$, single loci with even allele proportions, and placed randomly in a 20m x 20m area centred on (0,0), could be defined as:

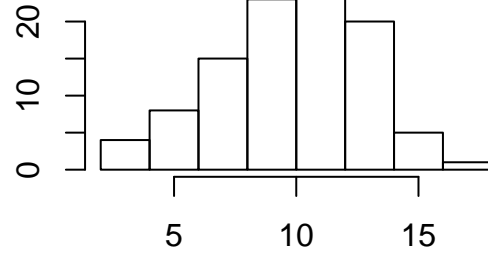
```
obs.win <- owin(xrange=c(-10,10), yrange=c(-10,10),unitname="metres")
curr.pop <- create.ibm.population(N=100,
                                spat.layout="random",
                                obs.win=obs.win,
                                prob.females=0.5,
                                age.distribution=c(10,3),
                                allele.prop=c(0.5))

par(mar=c(0,0,1,0)) # rescale figure border
plot(curr.pop,use.marks=FALSE,main="Initial Pop Example")
points(curr.pop,pch=21,bg=curr.pop$marks$sex) # Colour points
par(mar=c(3,3,2,0)) # rescale figure border
hist(curr.pop$marks$age,main="Age Distribution")
```

Initial Pop Example



Age Distribution



The variable `curr.pop` represents the initial population and is a point pattern of class “ppp”. The data for the population is held as a dataframe, accessible as `$marks`. The columns for the dataframe represent the unique id for each point, sex (1 = Male, 2 = Female), age in years, father (f) and mother (m) ids, and the list of allele values, 2 per loci. Note that the initial population has f and m set to zero, since there were no parents. For our example above there was only one locus, so we have just 2 allele values (V1 and V2). The first 5 points are shown in the table below.

id	sex	age	f	m	V1	V2
1	2	12	0	0	0	0
2	2	15	0	0	1	0
3	1	6	0	0	1	1
4	2	12	0	0	0	0
5	1	14	0	0	1	1

Typically the release window is smaller than the final window used for modelling movement and survival. Once the initial population has been created the final observation window is defined (in this case a 60mx60m window centred on the origin):

```
obs.win <- owin(xrange=c(-30,30), yrange=c(-30,30),unitname="metres")
curr.pop$window <- obs.win
curr.pop
#> Marked planar point pattern: 100 points
#> Mark variables: id, sex, age, f, m, V1, V2
#> window: rectangle = [-30, 30] x [-30, 30] metres
```

Behaviour

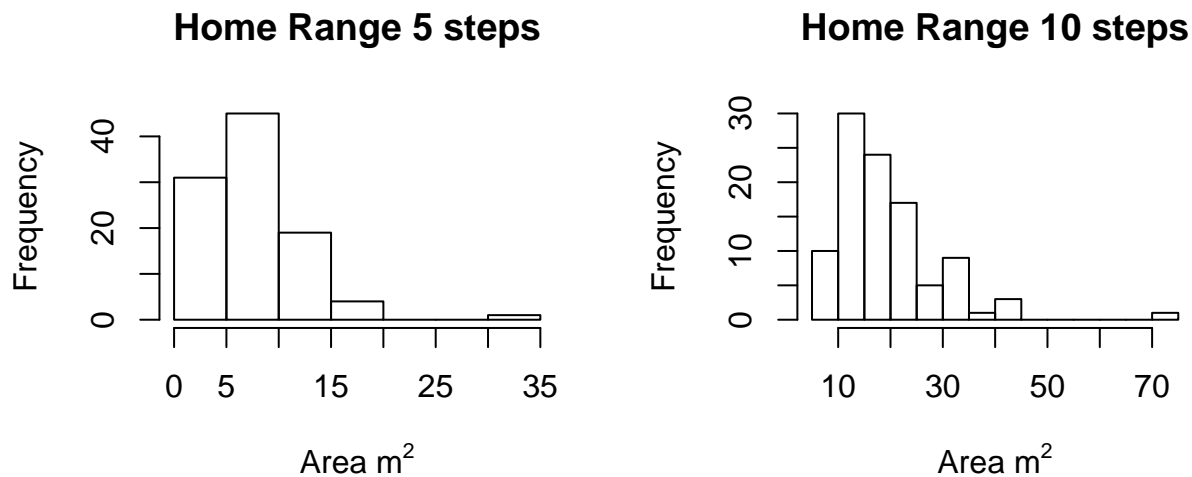
Movement

Movement is defined for different age classes and separately for males and females. Movement is assumed to be isotropic, with each movement sampled from a normal distribution $N(\mu, \sigma)$. The direction is random. For example, assume that males have a small movement until the age of 5, and then a larger movement, while females have a large movement irrespective of age:

```
m.age.class <- c(5,Inf)
m.move <- list(c(1,0.1),c(5,2))
f.age.class <- Inf
f.move <- list(c(5,2))
m.table <- movement.table(m.age.class,m.move,f.age.class,f.move)
```

The function `convex.home.range` allows an exploration of a movement parameter in relation to home range, estimated as the convex hull of movement for each individual. A histogram of areas can be used to help determine suitable parameters. For example, with $N(\mu=2, \sigma=1)$ over 5 generations (timesteps), the convex hull home range is:

```
m.age.class <- Inf
m.move <- list(c(2,1))
m.table <- movement.table(m.age.class,m.move,m.age.class,m.move)
areas <- convex.home.range(curr.pop,m.table,timesteps=5)
hist(areas,main="Home Range 5 steps",breaks=10,xlab=expression(paste("Area ",m^{2})))
areas <- convex.home.range(curr.pop,m.table,timesteps=10)
hist(areas,main="Home Range 10 steps",breaks=15,xlab=expression(paste("Area ",m^{2})))
```

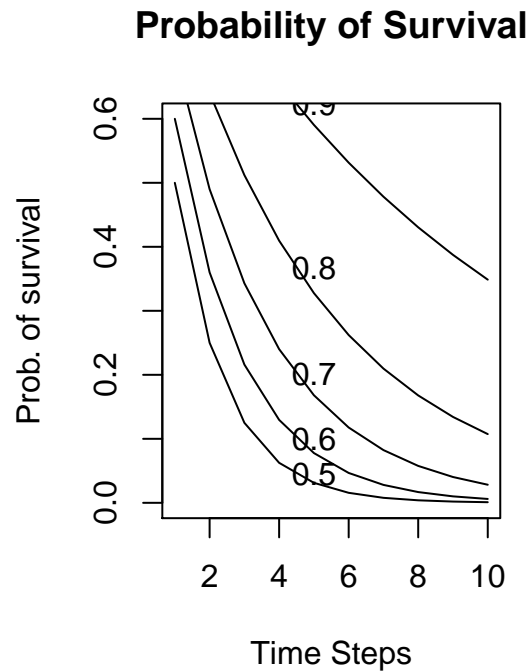
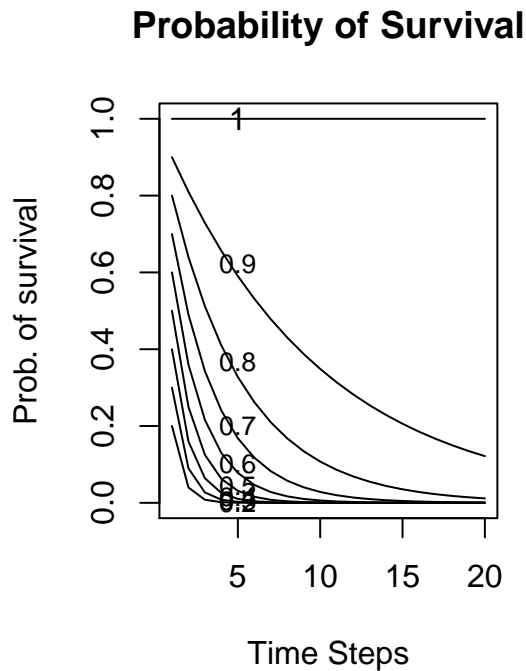


Survival Survival is defined for different age classes and males/females. The probability of survival for any age class is interpreted as the likelihood of survival to the next generation (timestep). Each age class may have a different probability. For example, if both males and females have a low probability of survival until the age of 5, a high probability of survival till 40, but very low survival (due to old age) after 40, the age and survive could be defined as:

```
mf.age.class <- c(5,40,Inf) # Same description for males and females
mf.survive <- list(0.8,0.95,0.1)
survive.table <- survival.table(mf.age.class,mf.survive,mf.age.class,mf.survive)
```

A simple function `plot.survival` is provided to explore how different probabilities effect the long term likelihood of survival.

```
plot.survival(p.seq=seq(from=1,to=0.2,by=-0.1),years=1:20,ylim=c(0,1),cex=0.8)
plot.survival(p.seq=seq(from=1,to=0.5,by=-0.1),years=1:10,ylim=c(0,0.6))
```



Breeding and Fecundity

Breeding occurs between a single male and female who are within a defined distance of each other. The probability of breeding depends on the age (which may be different for males and females). Given a breeding event, fecundity is drawn from a defined normal distribution to produce a random number of children. The children are randomly placed around the mid-point between the parents. Consider the following scenario: males and females must be at least 5 to reach maturity for breeding, then breed until age 40 after which time breeding is unlikely. They produce on average 5 children per breeding event, with children scattered on average about 3 metres apart in the first generation. The maximum distance between breeding pairs is 5 metres. This description is as follows:

```
#
# The breed.table and max.dist variable will be used in the model
#
mf.age.class <- c(5,40,Inf) # Same description for males and females
mf.breed <- c(0,0.95,0.1)
breed.table <- breeding.table(mf.age.class,mf.breed,
                             mf.age.class,mf.breed,
                             fecund=c(5,2), # Mean 5 children
                             displace=c(3,2)) # Offset from mid-point

max.dist <- 5.0 # Maximum distance between pairs
```

The following code shows the behaviour of placement of children. Here we define a population with 2 individuals, force them to be male and female with their midpoint at the origin, then produce some children and plot their position.

```
#
# The breed.table and max.dist variable will be used in the model
```

```

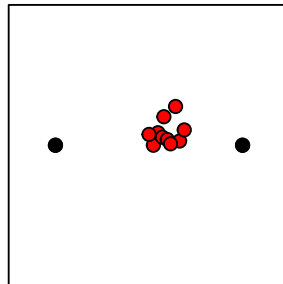
#
age.class <- Inf # For all ages
breed.always <- 1 # Definitely breed
b.table <- breeding.table(age.class,breed.always,
                          age.class,breed.always,
                          fecund=c(10,0), # Make 10 children
                          displace=c(5,1)) # Offset from mid-point
max.dist <- Inf # Any distance ok

# Make the population of 2 individuals
pop2 <- create.ibm.population(N=2,
                             spat.layout="random",
                             obs.win,
                             prob.females=0.5,
                             age.distribution=c(10,3),
                             allele.prop=c(0.5))

# Explicitly place the 2 individuals. This is not normally required.
pop2$x <- c(-20,20) # Either side of origin
pop2$y <- c(0,0) # On x-axis centred around origin
pop2$marks$sex <- c(1,2) # Make one male, one female
b.pairs <- breeding.pairs(pop2,max.dist)
children <- breed(pop2,b.pairs,b.table) # create children
par(mar=c(0,0,2,0))
plot(pop2,use.marks=FALSE,main="Children Placement")
points(pop2,pch=21,bg='black') # Parents Black
points(children,pch=21,bg='red') # Children Red

```

Children Placement



Crowding

The specification of crowding is used to control over-population in terms of population density within local neighbourhoods of each individual. Crowding can be used as a surrogate for nutrient requirements, density-dependent death and so on. Similar to previous definitions, age classes for each sex are defined, and for each age class probabilities can be given for a range of densities. Since this is a more complex definition than those for breeding and survival a function for visualising the crowding table has been provided (`plot.crowding.table`). As a simple example, assume that for both sexes different crowding is defined for age classes 1-5 and greater than 5. Crowding density less than 0.5 has no effect on survival, density 0.5 - 3 has a small effect, while greater than 3 has a large effect on survival.

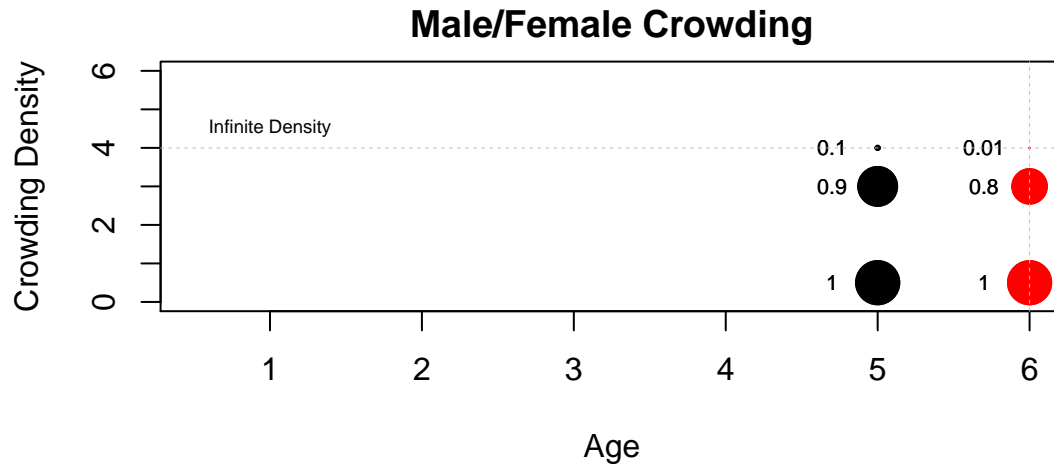
```

age.class <- c(5,Inf) # Same crowding for males and females

```

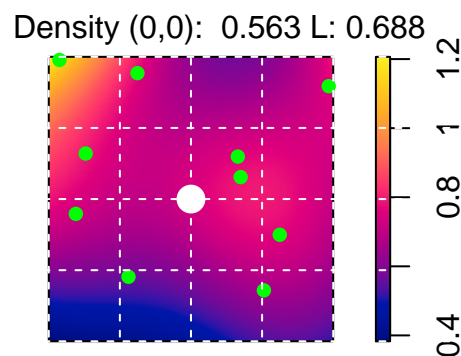
```
den.class <- c(0.5,3,Inf)
crowd.prob <- c(1,0.9,0.1,1,0.8,0.01) # age class x den class

c.table <- crowding.table(age.class,den.class,crowd.prob,
                          age.class,den.class,crowd.prob)
par(mar=c(4,4,2,0))
plot.crowding.table(c.table,main="Male/Female Crowding",sex=1)
```



A simple density plot is also supplied with spatibm that can be used to help visualise and therefore determine appropriate crowding measures. Below a density example is shown, with the heading for the plot giving the crowding density measure at the origin and the mean density for the surface (this is just the intensity, determined as number of points / area). Note that each cell in the plot is 1m x 1m.

```
par(mar=c(1,1,2,1.5))
# Create a window centred on (0,0)
win <- owin(xrange=c(-2,2),yrange=c(-2,2),unitname="metres")
plot.density.example(win,num.pts=10,sigma=1)
```

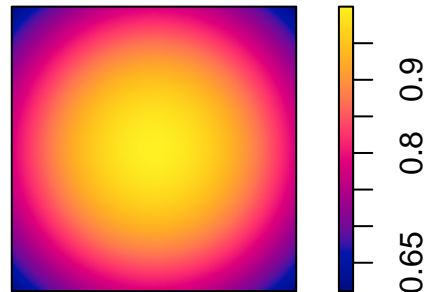


The Habitat Surface

The final aspect to survival is the influence of the habitat. This can be ignored, or used to represent the variable nature of conditions in the spatial window. The probability of survival for an individual is determined by the previous survival and crowding parameters, and then multiplied by the value at the habitat location where the individual is found. Note that this can be > 1 if a habitat can be considered to improve survival

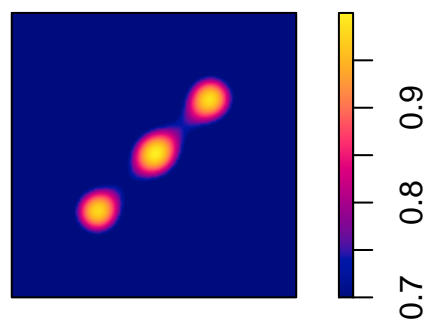
probability beyond a baseline measure. The surface uses a raster image and window (the owin assigned to pop\$window). A number of functions are supplied to create a habitat. The following produces a circular habitat, centred on the origin, with a value of 1 at the origin that decays slowly based on the distance decay parameter b.

```
par(mar=c(1,1,2,1.5))
hab.win <- owin(c(-50,50),c(-50,50),unitname="metres")
hab.surface <- as.im(circle.habitat,hab.win,b=0.0001)
plot(hab.surface,main="")
```



Sometimes you may want to examine several nearby good habitats that are separated by poorer habitat. The following example produces 3 locations that have good habitat (assumed to be the base level of 1.0), with the surrounding regions having a value of 0.7. Note the sigma and edge arguments passed to the density function.

```
par(mar=c(1,1,2,1.5))
hab.win <- owin(c(-50,50),c(-50,50),unitname="metres")
hab.pts.surface <- circle.pts.density(x=c(-20,0,20),y=c(-20,0,20),hab.win,
                                     min.d=0.7,sigma=10,edge=FALSE)
plot(hab.pts.surface,main="")
```



Putting it together

The model is run by producing the initial population, the tables for breeding, survival and crowding and the habitat surface. Breeding and survival must be defined while crowding behaviour and the habitat surface are optional.

```

# Create initial population
obs.win <- owin(xrange=c(-10,10), yrange=c(-10,10),unitname="metres")
curr.pop <- create.ibm.population(N=40,
                                spat.layout="random",
                                obs.win=obs.win,
                                prob.females=0.5,
                                age.distribution=c(5,3),
                                allele.prop=c(0.5))

# Expand to total area of habitat
curr.pop$window <- owin(xrange=c(-50,50),yrange=c(-50,50),unitname="metres")

# Define movement: same for males/females
age.class <- c(5,40,Inf) # Most active between ages >5 and <=40
move.all <- list(c(1,0.1),c(6,2),c(2,3))
m.table <- movement.table(age.class,move.all,age.class,move.all)

# Define survival: same for males/females
mf.age.class <- c(5,40,Inf) # Same description for males and females
mf.survive <- c(0.75,0.95,0.1) # Survive difficult to 5, then ok
s.table <- survival.table(mf.age.class,mf.survive,mf.age.class,mf.survive)

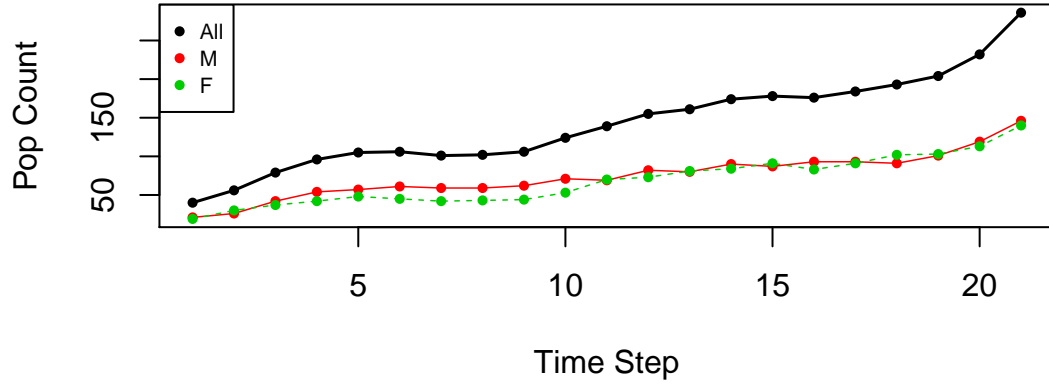
# Define breeding behaviour

age.class <- c(5,40,Inf) #
breed.probs <- c(0.0,0.8,0.01) # No breeding till >5 and <=40
b.table <- breeding.table(age.class,breed.probs,
                          age.class,breed.probs,
                          fecund=c(10,2), # Make about 10 children
                          displace=c(5,1)) # Offset from mid-point
max.dist <- 10.0 # Must be within 10 mtr

# No crowding or habitat surface.
# Run model for 20 generations

model.run <- multiple.step(
  steps=20,
  curr.pop,
  move.table=m.table,
  survive.table=s.table,
  breed.table,
  habitat.surface=hab.surface, # habitat defined earlier
  crowd.table=NA, # No crowding
  crowding.sigma=0.0,
  max.dist=max.dist, # Breeding maximum distance
  track.ids=FALSE,
  trace.output=TRUE)
#> Gen:1 N=40...Gen:2 N=56...Gen:3 N=79...Gen:4 N=96...Gen:5 N=105...Gen:6 N=106...Gen:7 N=101...Gen:8 N=100...Gen:9 N=101...Gen:10 N=102...Gen:11 N=139...Gen:12 N=155...Gen:13 N=161...Gen:14 N=174...Gen:15 N=178...Gen:16 N=176...Gen:17 N=175...Gen:18 N=174...Gen:19 N=173...Gen:20 N=172
plot.pop.count(model.run)

```

The result of running the model is a list of marked point patterns, one for each generation. A summary dataframe of the model is created using the `model.summary` function. These results give the demographic summaries for each generation, the spatial density properties based on sigma, and, for each locus, the allele frequencies and F statistics. Note that for our example there was only 1 locus, so there are just the 4 measures.

```
model.res <- model.summary(model.run,sigma=1,file.csv=NA)
```

time	n	mean.age	min.age	max.age	sexratio
0	40	5.675000	1	11	0.5250000
1	56	4.625000	1	12	0.4642857
2	79	3.924051	1	13	0.5316456
3	96	3.854167	1	14	0.5625000
4	105	3.933333	1	15	0.5428571

mean.den	min.den	max.den	pfreq1	Hobs1	Hexp1	Fstat1
0.0900865	1.73e-05	0.2168969	0.5625000	0.4750000	0.4921875	0.0349206
0.1544556	3.00e-07	0.4586604	0.5625000	0.4107143	0.4921875	0.1655329
0.1850187	0.00e+00	0.6076541	0.5443038	0.4810127	0.4960743	0.0303618
0.2081455	1.00e-07	0.7623016	0.5000000	0.4583333	0.5000000	0.0833333
0.2029448	0.00e+00	0.7097695	0.5000000	0.5047619	0.5000000	-0.0095238

Conclusion

This document has described the basic components of the `spatibm` package, but has ignored the use of habitat structures for modelling bottlenecks, or how more detailed concepts such as fixation of alleles, or the use of multiple runs for assessing population stability, can be performed. These concepts will be addressed in a second vignette.