Managing for ecological surprises in metapopulations

Supplemental materials

01 May 2019

## Metapopulation model

### Local & metapopulation dynamics

Our metapopulation is defined by a set of local populations with time-dynamics that follows birth (i.e., recruitment *R*), immigration, death, and emigration (BIDE) processes:

where is the number of adults in patch *i* at time *t*, is number of recruits, is number of recruits immigrating into patch *i* from any other patch, is number of recruits that die due to disturbance regime, is the number of recruits emigrating from patch *i* into any other patch, and is stochasticity in recruitment.

Resoure monitoring often occurs at the scale of the metapopulation, hence we define metapopulation adults as:

with metapopulation recruits:

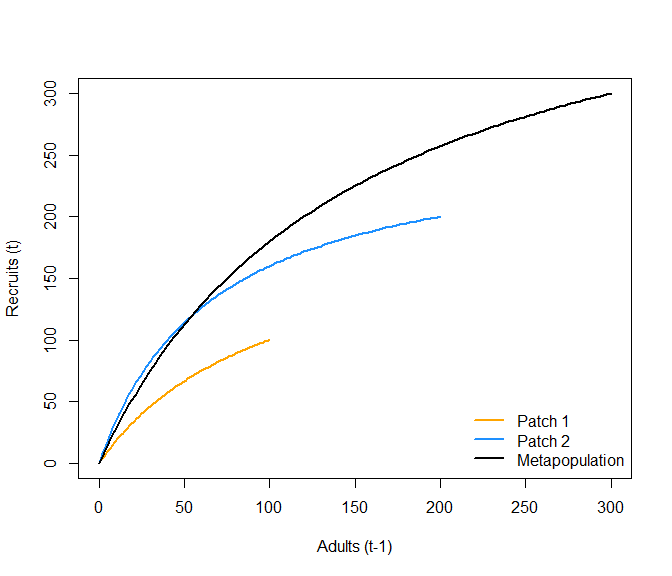
Local patch recruitment at time *t* depended on adult densities at *t-1* and followed a reparameterized Beverton-Holt function:

$R\_{it}=\cfrac{\alpha\_iN\_{it-1}}{1+\cfrac{\alpha\_i-1}{\beta\_i}N\_{it-1}}$

where is the recruitment compensation ratio and is local patch carrying capacity.

For example, in a two patch model that varies and parameters such that

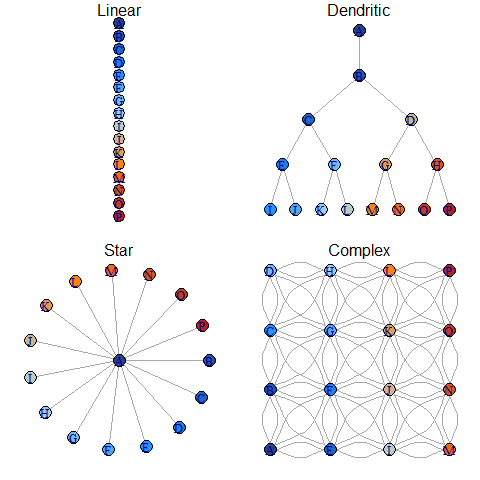
alpha <- c(2, 4)  
beta <- c(100, 200)

Management often monitors metapopulation resources as the aggregate of all local populations. In this way, recruitment compensation from local patches gets averaged across the metapopulation leading mean compensation of 3. Likewise, the total carrying capacity of the metapopulation becomes the summation of local patch carrying capacities , which is 300. This scale of monitoring generates the following local patch and metapopulation dynamics: 

### Creating the spatial networks

The next aspect to our metapopulation model is connecting the set of patches to one another. We need to specify the number of patches, their arrangements (i.e., connections), and how far apart they are from one another. We followed some classic metapopulation and source-sink arrangements to create four networks that generalize across a few real-world topologies: a linear habitat network (e.g., coastline), a dendritic or branching network (e.g., coastal rivers), a star network (e.g., mountain & valley), and a complex network (e.g., terrestrial plants).

To make networks comparable, each spatial network type needs the same leading parameters (e.g., and ) . In this case for number of patches, we set to 16 and to 1 unit (distance units are arbitrary). We used the igraph package and some custom code to arrange our spatial networks as the following:



Note that distances between each connecting patch (the links between nodes) in the above networks are equal.

An example dispersal matrix for the complex network:

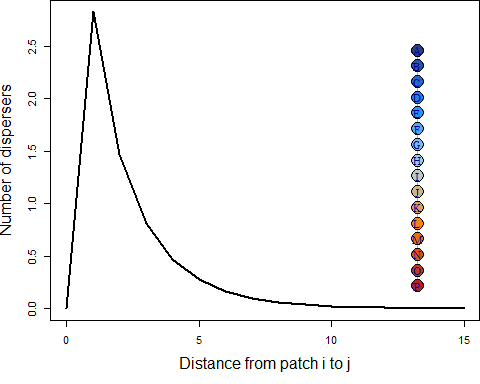
## A B E F C G D H I J K L M N O P  
## A 0 1 1 1 2 2 3 3 2 2 2 3 3 3 3 3  
## B 1 0 1 1 1 1 2 2 2 2 2 2 3 3 3 3  
## E 1 1 0 1 2 2 3 3 1 1 2 3 2 2 2 3  
## F 1 1 1 0 1 1 2 2 1 1 1 2 2 2 2 2  
## C 2 1 2 1 0 1 1 1 2 2 2 2 3 3 3 3  
## G 2 1 2 1 1 0 1 1 2 1 1 1 2 2 2 2  
## D 3 2 3 2 1 1 0 1 3 2 2 2 3 3 3 3  
## H 3 2 3 2 1 1 1 0 3 2 1 1 3 2 2 2  
## I 2 2 1 1 2 2 3 3 0 1 2 3 1 1 2 3  
## J 2 2 1 1 2 1 2 2 1 0 1 2 1 1 1 2  
## K 2 2 2 1 2 1 2 1 2 1 0 1 2 1 1 1  
## L 3 2 3 2 2 1 2 1 3 2 1 0 3 2 1 1  
## M 3 3 2 2 3 2 3 3 1 1 2 3 0 1 2 3  
## N 3 3 2 2 3 2 3 2 1 1 1 2 1 0 1 2  
## O 3 3 2 2 3 2 3 2 2 1 1 1 2 1 0 1  
## P 3 3 3 2 3 2 3 2 3 2 1 1 3 2 1 0

### Dispersal

Dispersal from patch *i* into patch *j* depends on constant dispersal rate (defined as the proportion of total local recruits that will disperse) and an exponential distance-decay function between *i* and *j* with distance cost to dispersal following:

with probability of dispersal from patch i into patch j:

where is the pairwise distance between patches and is the total dispersing animals from patch i into patch j. The summation term in the denominator normalizes the probability of moving to any patch to between 0 and 1. With , , , in a linear network:



### Recruitment stochasticity

### Disturbance

## Emergent outcomes

### Scale of management & monitoring

## Example scenarios

## 