

LAB EXERCISE #6 – Metapopulation Analysis

Instructors: E. Plunkett and K. McGarigal

Overview: In this exercise, you will learn to use a metapopulation viability model developed in R and interpret the output in the context of a case study on marbled salamanders in western Massachusetts. You will gain a practical understanding of metapopulation concepts in the context of a real-world application.

Objectives

To become more familiar with R and a matrix based population model.

To review and apply some basic concepts of metapopulation theory.

To compare multiple landscape scenarios and their consequences for metapopulation dynamics through a spatially explicit modeling approach based on the marbled salamander in western Massachusetts.

Background

Marbled Salamander Ecology.--The marbled salamander (*Ambystoma opacum*) is a terrestrial salamander whose natural range extends from the southeastern U.S. to southern New England. In Massachusetts, the species is listed as “Threatened” under the state Endangered Species Act. Here, its rarity can be attributed to both the northern climate and to habitat loss and fragmentation. To be successful, marbled salamander populations need both dependable, fish-free breeding habitats (i.e., vernal pools) and adequate terrestrial habitat for the remainder of their life cycle. Though little is known about their upland/terrestrial ecology, they are often associated with well-drained soils near moderate and sometimes rocky slopes.

Unlike other vernal pool amphibians in New England, the marbled salamander breeds in the late summer and early fall. After courtship, the females lay eggs in dry pond basins which soon fill with water, covering the eggs. The eggs hatch almost immediately, and if the larval salamanders are able to survive through the winter, they will metamorphose as the ponds begin to dry during the following spring and summer. Once in the terrestrial life stage, the animals reach sexual maturity in their second to fourth fall and return to the pond basins every one to three years to breed.

The Massachusetts climate keeps individual populations of marbled salamanders fairly small and subsequently more vulnerable to random fluctuations in climate or demographic success. The metamorph age class (from fall hatching to spring metamorphosis) is most vulnerable, and may often fail completely if a hard freeze precedes fall rains or if ponds dry in the spring before the larvae metamorphose. There is much less yearly variation in survival in survival of terrestrial age classes.

Metapopulations.--Metapopulations can be loosely defined as groups of discrete, local populations related by occasional dispersal events. Much theory has been developed on the

possible spatial dynamics of metapopulations, but there are very few empirical studies that adequately support or refute these theories. Vernal pool-breeding amphibians have been considered ideal study subjects for this purpose because they require discrete, ephemeral breeding habitats often embedded in a large woodland matrix. In addition, species such as the marbled salamander have been shown to have high breeding pond fidelity - that is, most metamorphs and adults return to their natal pond to breed. Only a small percentage of the youngest age classes attempt to disperse to other breeding sites. Known dispersal distances average approximately 200m with a maximum capacity of 1200m. Some logical questions that follow include:

Are individual “pond-populations” large enough to survive random environmental or demographic fluctuations?

Is some regional stability gained by the occasional interactions/dispersal events among local populations - allowing declining or extinct patches to be rescued or recolonized by other, more productive populations?

What conditions make these types of inter-population dynamics more or less likely, and what are the implications for the conservation of this species?

This exercise is drawn from a long-term study being conducted by the Department of Natural Resources Conservation in cooperation with the Massachusetts Natural Heritage and Endangered Species Program (MANHESP) to begin to address these questions.

Study Area Description.--The study site is on the south side of the Holyoke Range, situated mostly on the west side of the town line separating South Hadley and Granby, Massachusetts. This area is dominated by mature deciduous woodlands and occasional conifer stands on a rocky substrate. Steep, south-facing slopes characterize the northern portion, while topography through the remainder of the area is variable. One perennial stream (Elmer Brook) and several intermittent streams drain the area. Fourteen vernal pools have been identified within our 1-km radius sampling area – ten clustered tightly in the western section and four clustered loosely to the east. These vary in type, including scrub/shrub dominated, open-deep water, and open-shallow pools. As of November, 2006, twelve of these pools are known breeding localities for marbled salamander populations, ranging in size from 1 to 140 breeding females. Note, these populations appear to have uneven sex ratios, with many more males than females. For this reason, the model below is simplified to run for female abundances only - with the assumption that numbers of males will not be a limiting factor.

The core area is undeveloped, but contains many dirt roads and trails. These marked trails, and the area's proximity to the Holyoke Range State Park encourage many types of recreation, including hiking, mountain biking, ATV use, and hunting. Limited forestry has been practiced in a few areas, as evidenced by one clear-cut and another selective harvest. A power line corridor traverses the area laterally, and Rt. 116 is the only paved road, running along the southeastern perimeter of the site. Development in this area is currently concentrated along Rt. 116; however, increased pressures are mounting to open the area to residential development.

R is a statistical programming language and environment. It is open source, modular, powerful, flexible, and excels at producing graphics. It is not particularly friendly. The model

we're using is implemented almost entirely in R and has many of the same qualities. We'll mostly be interacting with the model by modifying its inputs files, but we will use R more directly to help us interpret and graph the results of the model runs.

Assignment

As a team, answer each of the following sets of questions:

- 1 Dispersal plays a fundamental role in metapopulation dynamics, acting as the only physical link among local populations. In a dynamic landscape, dispersal allows for vacated patches to be recolonized or for declining populations to be “rescued” by immigrants from more productive populations. In contrast, a population that is isolated by distance or development may be prone to extinction and have a low probability for recolonization due to low dispersal.

In this model dispersal rates are primarily determined by two parameters: *h.disp* and *disp.factor*. The first sets the standard deviation of the dispersal kernel the second the height. We have a good empirical estimate of *h.disp* but are less certain about the second parameter. Increasing this parameter can be thought of as increasing the success rate of individuals who choose to disperse.

To investigate the effect of varying dispersal on metapopulation dynamics compare the results of these four simulations: (1) *1a.no.dispersal*, (2) *1b.low.dispersal*, (3) *0.baseline*, (4) *1c.high.dispersal* representing zero, low, moderate, and high dispersal rates, respectively.

Using output files and plots of your choice, explain your results and the implications for metapopulation structure and viability.

- 2 A significant consideration in the investigation of metapopulation dynamics is the degree to which population fluctuations are correlated among sub-populations. Are sub-populations responding primarily to regional environmental variation with some degree of synchrony or are they subject to substantial local variation resulting from environmental or demographic factors? Compare the results of the following five simulations: (1) *2a.no.correlation*, (2) *2b.low.correlation*, (3) *0.baseline*, (4) *2c.high.correlation*, and (5) *2d.complete.correlation* reflecting increasing levels of correlation among local population dynamics (0, .3, .75, .9, and 1). Using any output files or plots you choose, explain the effects of varying degrees of correlation on a metapopulation and the reasoning behind your conclusions. Also discuss issues of scale involved with designating correlation among populations.

- 3 A local developer has chosen the study area as an ideal location for a golf course. “Salamander Links” will provide golfing in a natural setting with fabulous views of the Holyoke Range. Existing seasonal ponds will be utilized as water traps in the course.

You have been hired by the developer to use your understanding of salamander ecology and metapopulation dynamics to minimize the impact of the development on the amphibians. However, the developer insists that at least 3 of the pools must be incorporated into the fairways as water traps. Furthermore they intend to build a clubhouse on the high ground in

the northeast corner of the site (above 403 ft elevation), which will be accessed by a road running through the property either to route 116 or to the side road in the southeast corner.

Working within these constraints layout a golf course and use the model to assess its impact on the salamander population (See Appendix F)

Discuss your findings by comparing the costs and benefits of alternative scenarios. Discuss the advantages and/or risks of using this specific modeling approach to conservation planning.

- 4 Any model is dependent on its assumptions. Appendix C is a list of the models parameters, the value used in our baseline run, and a completely subjective estimate of confidence in that parameter.

Built into this modeling framework is the ability to estimate the sensitivity of the model outcome to variations in each parameter. This is done by sampling parameters for each run from a normal distribution centered on the specified parameter value - essentially varying each parameter slightly between runs. Importantly the sensitivity of each parameter depends on the specifics of the simulation: the value of each parameter, and the spatial arrangements of pools and land cover. For example the sensitivity of the model to changes in dispersal parameters may be low if there is only one pool and high when there are many.

We can analyze the sensitivity of any model configuration by setting `ignore.cv` set to `FALSE` in the batch file and increasing the number of reps (at least 1000 are needed). We've done this for the baseline scenario with the `4a.baseline.sensitivity` run. Plot the results using the `plot.error.analysis()` function. What parameters is the model most sensitive to? Do any values surprise you? Given the confidence in each parameter listed in Appendix C and the sensitivity analysis which parameter do you think most undermines the confidence we can place in the results? How does scale play into that uncertainty?

If time allows conduct sensitivity analysis of your development scenario from 4. How has the sensitivity of the parameters changed from the baseline run? How does the sensitivity analysis inform your confidence of the assessed impact of development?

General Guidelines

Each team will be assigned a single question to present orally, and will be responsible for submitting a written report on the remaining questions. The report should be well organized and presented in a clear and concise manner. There is no minimum or maximum page limit, although you should be able to complete the assignment in less than 4-5 single-spaced pages. Full references should be included for all literature cited. Papers should be submitted electronically. Reports should demonstrate that an interdisciplinary approach was used. In other words, I want a single integrated presentation that effectively demonstrates that the entire group discussed the results and reached some kind of consensus over the interpretation of those results. The report should not consist of several separate efforts combined at the end for purposes of the presentation. Moreover, the concepts discussed in lecture should be integrated into the discussion of the results to the extent possible.

One final question to ponder (but you don't need to report on): several variations on metapopulation theory were introduced in lecture, including mainland/island (core/satellite) dynamics, source-sink dynamics, patchy populations, non-equilibrium populations, and classic metapopulations. Which, if any, of these variations best describe dynamics in our baseline simulations of the marbled salamander population?

APPENDICES

A. Model Use

Housekeeping

The entire model directory should be moved and/or copied as a unit. Once it's in place the third line of two files must be adjusted to point to that directory. The files are: *model/code/main.r* and *model/code/analysis/setup.analysis.r* the third line should be set to something like:

```
my.wd <- "Z:/Ethan/Model/"
```

Note those are forward slashes not back slashes.

Mac Users: change the seventh line instead.

Running a batch

- 1 Open the file *model/code/main.r* and source it by selecting "Run all" from the "edit" menu.
- 2 Switch to the console window and type: `run.batch()` which will run the model using the default batch file: *input/batch.csv*.
- 3 To speed things up repeat steps one and two on as many computers as you can.

Note: If you'd like to specify a different batch file you can by calling `run.batch()` with the path of that file. For example `run.batch("input/batch2.csv")`.

Setting up a batch

There are a bunch of files within the *input/* directory used to configure the model.

parameters.csv - Each column represents a parameter. Each line represents a different set of parameters which can be specified by the batch file.

scenarios.csv - Each line specifies a spatial aspect of the input by referencing an arrangement of pools and potentially an input land cover map. It also allows for configuration of different forestry treatments for the matrix and buffers around pools, which we won't be doing in this lab.

arrangements/ Here's where we specify the locations, sizes, initial population size, and hydro periods of pools. We can open up *arrangements/1.csv* to see what we'll be using for all of our model runs.

batch.csv Is where we actually tell the model what we want to run. Each line represents a simulation and the columns specify aspects of that simulation:

scenario – the id number of the scenario we want to use

parameter.set - the parameter set we want to use

n.reps - how many reps (100 except for error analysis where we need 1000)

out.file - the name of this simulation it will be used to name the output. Note: the model will run fine if two lines in the batch file have the same *out.file* specified. However, the results of the two runs will be appended to the same file likely creating a mess. Make sure each line has it's own *out.file*.

function.call – which function to use to run the simulation we'll always be using "static.run". Other functions that I've written include: "dynamic.run" (for when the landscape changes during the course of the run), and "dispersal.run" (for keeping track of

the overall dispersal rate for model calibration).

ignore.cv - if TRUE the model will set the parameters precisely at their given value. If FALSE the model will vary the parameters around the specified value randomly (based on the *cv* specified for each parameter).

return.raw.data – we’ll always have this set to TRUE

current.rep – initially this should be 1. As the batch is run it gets updated. Note, this column isn’t necessary; it will be created during the run if it doesn’t exist.

Viewing the Results:

The model results will be saved to the *Model/output/* directory. There should be a directory for each line of the batch file. Within each folder there will be two or three files:

raw.data.csv – the entire population vector for each time step.

summary.stats.csv - summary stats on each run

parameter.states.csv – This is only output if *ignore.cv* is set to FALSE in which case it stores the specific parameter states for each run of the model.

You’re welcome to look at and analyze these files in excel or with any other means you choose. However, I’ve also written a small collection of R functions to graphically display the data in these files. To load these functions into R open up the *code/analysis/setup.analysis.r* file in R and then select “source file” from the “file” menu. Functions I’ve written for you are:

```
plot.error.analysis()  
plot.extinction.risk()  
plot.metapop.size()  
plot.populations()
```

All of these functions take as their first parameter a simulation name or a vector (or list) of simulation names. However, this parameter is optional and if left out the functions will attempt to plot graphs for all of the model output in the *output/* directory.

If in the course of working on this lab you decide you’d like to have some other plot of the data please let me know and I’ll try and write a function to meet your needs.

Appendix B – Sample code for plotting results:

```
# make vectors of run names (for convenience)
# ( doing this allows us to substitute the text to the left of
# the <- operator for the text on the right in function calls)
d.runs <- c("1a.no.dispersal", "1b.low.dispersal", "0.baseline",
"1c.high.dispersal")
c.runs <- c("2a.no.correlation", "2b.low.correlation", "0.baseline",
"2c.high.correlation", "2d.complete.correlation")

# Example Plots
# You will probably want to make additional plots by calling
# these functions with different run names. Calling
# any of these function with no parameters will generate graphs
# for all simulation runs.

# plot the total metapopulation size over time for all reps of the
# baseline run
plot.metapop.size("0.baseline")

# generate the same plot for every simulation one by one
plot.metapop.size()

# plot each populations' trajectory rep by rep for the baseline scenario
plot.populations("0.baseline")

# plot the extinction.risk of a single run
plot.extinction.risk("0.baseline")

# plot the extinction risk of all the dispersal runs on one graph
# (the following two lines produce identical graphs)
plot.extinction.risk( c("1a.no.dispersal", "1b.low.dispersal", "0.baseline",
"1c.high.dispersal"), ylim=c(0,30))
plot.extinction.risk(d.runs, ylim=c(0,30))

# plot the extinciton risk for all of the catastrophe runs:
plot.extinction.risk(c.runs, ylim=c(0,30))

# sensitivity plot for the baseline scenario:
plot.error.analysis("4a.baseline.sensitivity") # 100 year extinction risk as
response variable (default)
plot.error.analysis("4a.baseline.sensitivity", y="mean.pop.size") # mean
metapopulation size as response variable
plot.error.analysis("4a.baseline.sensitivity", y="turnover.rate") #
turnover.rate as response variable
plot.error.analysis("4a.baseline.sensitivity", y="mean.patch.occupancy") #
mean patch occupancy as response variable
```

Appendix C - Model Parameters

	Value	Confidence (subjective)
Spatial Parameters		
canopy.threshold	85	10
max.forest.resistance	3	4
h.disp	440	8
disp.factor	32	4
h.surv	103	8
min.adult.surv	0.4	3
min.j.to.a.surv	0.025	3
watershed.r	100	6
max.d.h.p	17	6
Metapopulation Parameters		
adult.surv	0.61	9
j.to.a.surv	0.106	9
fec.cor	0.76	7
optimal.hp	256	8
hp.fec.v.scale	1255	7
hp.fec.sd	29	8
max.scaled.fec	1.64	7
cat.rate	0.61	7
cat.cutoff	0.1	7
allee	1	3
breeding.rate	0.644	9
density.fun	taylor.scott.1997	8
dp1	1	5
dp2	NA	NA
dp3	NA	NA
Implementation and Model Structure		
run.length	100	NA
cell.size	10	NA
n.stages	4	NA
min.lc.range	1	NA
max.lc.range	200	NA
min.t.range	201	NA
max.t.range	300	NA
min.c.c.range	400	NA
max.c.c.range	500	NA
forest.lc.code	2	NA

Appendix D – A brief descriptions of parameters

Spatial Parameters

canopy.threshold, *min.forest.surv*, *watershed.r*, *max.d.h.p*, and *max.forest.resistance* are all used to relate different degrees of timber harvest to dispersal and survival of salamanders. They have no effect on runs done in this lab where all forest is treated the same.

h.disp is the standard deviation in the dispersal kernel in a minimally resistant environment (forest). Increasing it increases the distances animals are able to disperse.

disp.factor Set's the height of the dispersal kernel. Increasing it will increase the number of effective dispersers.

h.surv The standard deviation in the upland habitat kernel. This defines the area around the pool that animals use as habitat, the quality of which impacts survival of upland stages.

min.adult.surv and *min.j.to.a.surv* set the lower limit on survival for each of those stages, which would be achieved if the volume of the habitat kernel is 0. Essentially lowering these values increases the impact of habitat loss on each of those stages.

Metapopulation Parameters

adult.surv sets the survival rate of adults if the upland habitat is undisturbed.

j.to.a.surv sets the survival rate of juveniles if the upland habitat is undisturbed.

fec.cor is the correlation in fecundity (metamorphs/breeding female) across ponds. The higher it is the more likely all ponds will perform similarly in a given year.

optimal.hp, and *hp.fec.sd* define the relationship between a pond's hydro period and the maximum fecundity that can be achieved at that pond.*

hp.fec.v.scale, and *max.scaled.fec* increasing either of these will increase fecundity in non-catastrophe years. *

cat.rate The percentage of catastrophes (breeding failures). *

cat.cutoff The maximum fecundity (prior to scaling for the hydroperiod effect) of pools experiencing a catastrophe. Increasing this will marginally increase the fecundities of pools experiencing a catastrophe.

allee An allee effect is a depression in vital rates when the population gets very low. Think of it as a cost of finding a mate when there are only a few animals in the population. In this fecundities are multiplied by $N/(N+A)$; A is the *allee* parameter and N is the number of breeding females.

breeding.rate The percentage of females that attempt to breed in any given year.

* Note: The function that generates fecundities is complicated. It stochastically generates fecundities incorporating the covariance in fecundities, the relationship between hydroperiod and peak fecundity, and the catastrophe rate. If you really want to know more about how it works ask me, but I'd advice not getting too bogged down in the details.

Appendix E - Building a land-cover map to input into the model.

To answer question 3 we'll have to provide the model with a land-cover map in ascii grid format.

Follow these steps to build that map:

1. Make sure that you're working in a copy of the *Model* directory that belongs to your group.
2. Open the *fieldsite.mxd* document within the *Model/arc* directory.
3. Within arcmap turn on the editor toolbar and "start editing".
4. Select "Fairways" as the target layer and use the edit tool (the arrow on the left of the editor toolbar) to move the fairways around. Recall that 3 of them have to overlap vernal pools.
5. Switch the target layer to "minorroads".
6. Select the pencil tool, make sure the "create new feature" task is selected, and click points to layout your road staying within the parcel and connecting one of the other roads on the map to the higher ground in the northwest of the site.
7. Select save edits from the Editor menu.
8. Run arcinfo workstation (Start, All Programs, ArcGIS, Arcinfo Workstation, Arc).
9. Type something that looks kind of like the following but points to your folder instead of mine: `w C:\Model\arc\` (and then hit enter)
10. Type: `&r makegrid` (and then hit enter)
11. After a bunch of churning you should end up with some new files in the *Model/arc* directory: *result.asc* is the one we need. Rename it something appropriate and unique to your group and then copy it into: *Model/input/maps/landcover*. There should also be two new folders: *info* and *results*. *results* is a coverage that you can load it into arcmap if you want to see what the grid looks like.

Note: if you want to repeat this process you must delete or rename all of the output from the makegrid command: *result.asc*, *result*, and *info*.

Good job! Now we need to setup and run the model.

1. First we need to create a scenario with your input map. Open up *Model/input/scenarios.csv* and copy the last line and paste it as a new line. Update the *scenario.id* field (it must be unique) and the *input.lc.map* field (it should be the same as your map name including the extension).
2. Update the batch file.
 - a. Copy the first line of the batch file and paste it onto the first empty line.
 - b. Set the scenario for that line to match the id of the scenario you just created.
 - c. Set the out.file of your new batch to something unique to your group if you want to keep to our naming convention you should start with "3."
 - d. Set the current.rep of you new batch line to 1 (The current.rep of all prior lines should be greater than the n.reps for those lines.)
 - e. You may want to add an additional line to the batch file for a sensitivity analysis of this development run. It will be similar to the line you just created except have a different name, have ignore.cv set to FALSE, and have at least a 1000 reps (under n.reps).
3. Run the model. (Follow directions in Appendix A)