

Tutorial For Running R Code

Tristan Skraber and Joanna A. Bieri

Contents

1	Introduction to Program	1
1.1	Downloading and Preparing the Code	1
1.2	How To Tell If It's Running	1
1.3	Basic Overview	2
2	Specific Example	3
2.1	How to change Baseline Parameter Values	3
2.1.1	Survival Rate	3
2.1.2	Reproduction Rate	4
2.1.3	Migratory Probabilities and Survival	5
2.2	How to change Density Dependent Functions	5
3	Troubleshooting	6
3.1	General Comments	6
3.2	Possible Errors	6
3.3	Known Mac Specific Errors	6

1 Introduction to Program

1.1 Downloading and Preparing the Code

- Download the code: **NIMBioS-NetworkCode-MigrationNetwork**
- This folder will contain different example simulations:
 - FinalCodeElk
 - FinalCodeExample
 - FinalCodeMonarchs
 - FinalCodePintails
 - FinalCodePlants

along with the Network code 1.1

- The Network code 1.1 will not need to be changed by the user. It is called on by the species code so that it can run the simulation.

1.2 How To Tell If It's Running

- Open the folder titled **FinalCodeExample**.
- In this folder open the code titled **ExampleSimulation.R**. This is the main code that runs the example simulation. Source this file to run the simulation.

- To tell if the code is running correctly, the user should see 2 graphs pop up along with the predicted Equilibrium Population for each season printed in the console.
- If no errors occur, the first output, Equilibrium Populations for season 1 and 2, should read 0.009104905 & 0.007374973 respectively.
- Also printed to the console are Node Equilibrium Population Distribution and Path Equilibrium Population Distribution.
- If any errors are shown, refer to the **Troubleshooting** section for possible solutions.
- Two files are saved in the folder **Baseline1**: The full simulation in the Baseline1.RData file and the equilibrium population data in SteadyStatePopulation.csv.

1.3 Basic Overview

- This code is to help model migration and survival for species specific models that match the general network framework as found in the paper:

A general modeling framework for describing spatio-temporal population dynamics.

by Christine Sample* , John Fryxell, Joanna Bieri, Paula Federico, Julia Earl, Ruscena Wiederholt, Tyler Flockhart, Sam Nicol, Jay E. Diffendorfer, Brady J. Mattsson, Wayne E. Thogmartin, Richard A. Erickson, and D. Ryan Norris.

- For each species there are the following provided files:
 - **<species>Simulation.R** - this is the main code that is sourced to run the simulation.
 - SpeciesFunctions.R - this file contains the definitions for the density dependent functions used in the network simulation.
 - Baseline1 - this folder contains .xlsx spreadsheets with species and class specific data. This includes initial populations, and season specific survival and transition rates.
- With in each file there are multiple comments that explain variable names and important distinctions. Please read the comments to see specifics of each part of the code.
- NetworkCode1.1 contains the general network code that is called by each of the species simulations. The following files are provided:
 - NetworkSimulation.R - the full simulation calculation, time stepping, and convergence.
 - NetworkSetup.R - reads in the .xlsx files provided, creates variables, and set values for all network parameters to initialize the code.
 - NetworkOutputs.R - creates graphs and tables for automatically generated output variables.
- Important variables:
 - $N[[t]]\$<class>$ contains the population at time t for the specified class at each of the nodes.
 - $M[[t]]\$<class>$ contains the population at time t for the specified class along each of the paths between the nodes.
 - *timestep* indicates which time step in the simulation corresponds to season 1 in the converged code. So for example when running the three season elk simulation *timestep* = 27 which means that to see the equilibrium population at the nodes for one full annual cycle you would consider $N[[27]]$ = season 1 population, $N[[25]]$ = season 2 population, and $N[[26]]$ = season 3 population.

2 Specific Example

Here we will outline how to run and change values for one the Elk example. All of the example species follow the same general format.

2.1 How to change Baseline Parameter Values

- The Baseline1/spreadsheets are where many parameters can be modified to view the effect of changing constant rates of survival or path transition. To show how to modify these changes, we are going to use the Elk adult simulation as an example.
- To access the Excel file, go into the folder titled **FinalCodeElk**. Then open **Baseline 1**.
- Open the file named **network_inputs_adults.xlsx**.
(Note: This is only the parameters regarding the adult population of Elk)
- Within the file are tabs for **Initial Pop**, **Winter/Spring** and **Summer/Fall**.
- Variables are stored in a specific order:
 - Initial Pop Stores data about the initial population for each node.
 - All additional tabs should reflect seasons, and be ordered sequentially.
 - Under each seasonal tab:
 - * **NODE SPECIFIC:** properties are at the top of the sheet - variables names are taken from the first row and the word STOP tells the code when you are done defining node parameters
 - * **EDGE TRANSITIONS:** edge transition probabilities are listed below node specific properties leaving one blank row. For each node with the source listed on the first column and the destination in the first row. The rows must sum to one, to reflect the probability that all animals transition to somewhere.
 - * **EDGE SURVIVAL:** edge survival rates are listed below edge transitions leaving one blank row. For each node with the source listed on the first column and the destination in the first row.

2.1.1 Survival Rate

- We begin by looking at the **Winter/Spring** tab.

Node attributes		Adult survival	Reproduction	Carrying Capacity	Tell code to STOP
ALPHA		s0	r	K	STOP
Variable Names					
	1	0	0	1,640	
	2	1	0	1427	
	3	1	0	2172.7223	
Transitions pij		Destination			
	1	0	2	3	
	2	0	0	0	
Origin	2	1	0	0	
	3	0.5	0	0.5	
Edge survival sij		Destination			
	1	0	2	3	
	2	0	0	0	
Origin	2	1	0	0	
	3	1	0	1	
s33*p33_t-1		Destination			
	1	1	2	3	
	2	1	1	1	
Origin	2	1	1	1	
	3	1	1	1	

- To change the survival rates we must look at the numbers under **Adult survival**.

- The numbers 0, 1 and 1 correspond to the survival probabilities of the Elk during the Winter/Spring season for nodes 1, 2 and 3 respectively.
- Increasing these numbers would represent increasing the survival probabilities of the Elk for that node.

(Note: The survival rate for node 1 is 0. This is because during the Winter/Spring season, there are no Elk in that node.)

2.1.2 Reproduction Rate

- The reproduction rates are found in the column next to the survival rates:

Reproduction	
r	
	0
	0
	0
	0

- When changing the reproduction rates they represent:
 - 0 - no new animals are being produced
 - 0.5 - every two animals are producing a new animal
 - 1 - every animal is producing a new animal

(Note: Reproduction at each node is 0 because during the Winter/Spring season there are no new animal being born since it is not their breeding season)

2.1.3 Migratory Probabilities and Survival

		Destination		
Transitions pij		1	2	3
Origin	1	0	0	0
	2	1	0	0
	3	0.5	0	0.5
		Destination		
Edge survival sij		1	2	3
Origin	1	0	0	0
	2	1	0	0
	3	1	0	1

- On the left is the **origin** or the node the elk are leaving, and at the top is the **destination** or the node they are traveling to.
- Make sure the transition rate leaving each node sums to 1 because all the surviving animals must travel somewhere (or stay at that node).

3	0.5	0	0.5
----------	------------	----------	------------

(Note: This is the probabilities for which node the Elk will travel to when leaving node 3. Notice how $0.5+0+0.5=1$.)

- Survival rates can be changed in anyway, however numbers greater than one would imply reproduction along the path.

2.2 How to change Density Dependent Functions

- Species specific density dependent functions can be found in the file **SpeciesFunctions.R**. The simulation calls this file at each time step and recalculates values based on current population and parameter numbers.
- This file contains three functions:
 - **f_function** - the node update function updates the population at the nodes. Here we would define density dependent survival, reproduction, and class transition rates.
 - **p_function** - the transition probability function updates transition probabilities between the nodes. Here would would define density or temporally dependent transition functions.
 - **s_function** - the edge survival function updated survival between the nodes. Here we would define density or temporally dependent edge survival functions.
- Each function has a description, commented at the top, of the types of inputs that may be included in the function.

3 Troubleshooting

3.1 General Comments

- File structure is important! The contents of this folders must stay where they are. If items are moved around, the code will not be able to call upon them to run the simulation. If you change the file structure you will need to make corresponding changes in the code.
- Be sure to keep track of how things are labeled because the folder structure is sensitive. All file names need to match what is defined in the `<species>Simulation.R` code. For example:
 - `SIMNAME` - gives the exact folder name where the spreadsheets for the simulation parameters can be found. If running multiple perturbations, you can change the folder names here.
 - `NETNAME` - gives the exact names of the network classes. For example for the elk this is "adults" and "calves". This must exactly match the names of the .xlsx input files.

3.2 Possible Errors

- Always check for convergence!! Occasionally the code might run and produce numbers, but these will be incorrect due to an improper time step size. The code will output text that says **The simulation did not converge within the maximum time allowed** . There are a few possible solutions:
 - Increase the number of time steps to allow the solution to converge. Increase `tmax` in the code.
 - Increase the error tolerance. Convergence is assumed when the population of each classes changes less than **ERR** over one annual cycle. For our simulations we choose `ERR<.01`, but this may be more or less restrictive than your case requires.
- Some users might not have the right packages downloaded and this will produce an error so make sure all required packages are downloaded. This code requires the library(XLConnect) package.

3.3 Known Mac Specific Errors

- If code is not running for Mac users, make sure **XQuartz** is downloaded and running.
- Some users might see an error when using **rJava** in our experience this has been fixed by creating symbolic links to your working java libraries. For example:

```
sudo ln -f -s $(/usr/libexec/java_home)/jre/lib/server/libjvm.dylib /usr/local/lib
```