

ssmc User Guide

Bill Peterman

2015-09-29

ssmc: An R Package for assessing source-sink population status and contribution of populations to the metapopulation using demographic connectivity networks and Monte Carlo simulation

Background

The functions found in this package were originally developed as part of research focused on understanding source-sink populations dynamics of amphibians. This research was supported by the Department of Defense Strategic Environmental Research Development Program ([SERDP](#)). This document is meant to serve as a supplement to the documentation found in the `help` files of `ssmc`. The math behind the model is presented below, but understanding the math is not essential to applying and the using the model. Please carefully read this User Guide as well as all documentation related to the functions in the R package.

Modeling process

There 5 steps in the modeling process:

1. Select values for parameters
 - population size – assessed separately for each population
 - size of metamorphs⁺ – assessed separately for each population
 - proportion of population that is philopatric – equivalent among all populations at each iteration
 - probability of surviving to adulthood⁺ – equivalent among all populations at each iteration
 - dispersal distance – equivalent among all populations at each iteration
2. Determine probability of dispersal (see ‘Calculating dispersal’ below)
3. Determine the number of individuals dispersing between populations
4. Determine the contribution and influence of each pond to the metapopulation
5. Repeat steps 1–4 for the specified number of Monte Carlo iterations

Each of the above stochastic variables is drawn from a normal distribution with a mean and standard deviation that is specified based on your system. At each Monte Carlo iteration, different values are drawn.

After all iterations are complete, the ratio of immigrant:philopatric ($\frac{i}{p}$) individuals is assessed. If $\frac{i}{p} > 1$, the population is considered a sink, if $\frac{i}{p} \leq 1$ the population is considered a source. This evaluation is made for each iteration, and the frequency at which a population acts as a source or sink is evaluated. Additionally, the average number of emigrants, immigrants, and philopatric individuals is determined for each population, as well as the average contribution of each population to the metapopulation.

⁺ The size of metamorphs, if specified, is used to determine the population-specific probability of survival to adulthood using the equation:

$$\text{logit}(\text{survival}) = -1.366 + 0.87 * \text{size},$$

If metamorph size is not provided, then survival probability is drawn from distribution and will be uniform across all populations. The above equation come from Altwegg & Reyer ([2003](#))

Calculating dispersal

The probability of successful dispersal from population i to population j is calculated as:

$$m_{ij} = \exp(-\alpha d_{ij}),$$

for $j \neq i$ and $m_{ii} = 0$. Average dispersal distance is $1/\alpha$ and d_{ij} is the distance between populations i and j . The probabilities of m_{ij} are standardized so that rows i sum to one by

$$m'_{ij} = \frac{m_{ij}}{\sum_j m_{ij}}.$$

The number of individuals, n , dispersing from population i to j is then determined using a multinomial distribution such that

$$P(n_1, n_2, \dots, n_k) = \frac{N!}{(n_1! n_2! \dots n_k!)} * (p_1^{n_1} p_2^{n_2} \dots p_k^{n_k})$$

where

- N is the number of emigrants from i
- k is the total number of populations that individuals could disperse to
- n_{ij} is the number of dispersers from population i to j
- p_{ij} is the probability dispersing from population i to j , as determined from m'_{ij}

Values n_{ij} are components of the landscape matrix M_{ij} . Summing across columns, j , of matrix \mathbf{M} gives the total number of emigrants leaving population i , while summing across rows, i , gives the total number of immigrants entering population j .

$$\begin{aligned} emigrants_i &= \sum_{n=1}^j M_{ij} \\ immigrants_j &= \sum_{n=1}^i M_{ij} \end{aligned}$$

Determining contribution to the metapopulation

Metapopulation capacity, as defined by Hanski & Ovaskainen (2000), is a measure that encompasses the arrangement of habitat patches as well as the size of those patches. Size is used as a proxy for population carrying capacity, which is replaced with estimated abundance in the model formulation of `ssmc`. Following Hanski & Ovaskainen (2000), we assess the metapopulation capacity, λ_M , which is the leading eigen-value of landscape matrix \mathbf{M} . The individual contributions of each population, i , to λ_M is

$$\lambda_i \equiv x_i^2 \lambda_M,$$

where x_i is the i th element in the leading eigenvector of the landscape matrix \mathbf{M} . As such, λ_i measures the contribution of each population to metapopulation capacity. The importance of population i can be further assessed by removing it from the metapopulation network and recalculating λ_M . Alternatively, it is possible to assess the contribution of newly created habitat to the metapopulation by adding new population i to the network. This is what the functions `site.analysis` and `best.locale` of the R package `ssmc` seek to do.

Setup

Install necessary software and packages You will also need to have [R >= v3.0](#) installed. I would highly recommend installing [R studio](#) when working with R.

Getting the R Package

First, install `ssmc` from GitHub. This will require the `devtools` package

```
# Install 'devtools' package, if needed
if(!("devtools" %in% list.files(.libPaths()))){
  install.packages("devtools",
    repo = "http://cran.rstudio.com",
    dep = TRUE)
}

library(devtools) # Loads devtools

install_github("wpeterman/ssmc") # Download package
```

Load `ssmc` and clear your workspace.

```
library(ssmc)
rm(list = ls())
```

Assessing the source-sink status and importance of populations to the metapopulation The `ssmc` package comes with a simulated sample data set called `site.dat`. After loading the `ssmc` package, you can explore the structure and format of the data file.

```
str(site.dat)

## 'data.frame': 25 obs. of 7 variables:
## $ site : int 1 2 3 4 5 6 7 8 9 10 ...
## $ x : num 25.1 93.5 32.3 45.2 96.2 ...
## $ y : num 0.435 0.783 0.192 0.301 0.572 ...
## $ ab.mean : num 0 76.5 0 204.4 241.3 ...
## $ ab.sd : num 0 5.4825 0 1.5547 0.0809 ...
## $ svl.mean: num 1.057 0.4552 -0.4095 0.0623 -0.9316 ...
## $ svl.sd : num 0.369 1.363 1.402 0.721 0.802 ...
```

There are seven variables included in the data frame. This format is not essential to using the functions of `ssmc`, as parameter values are individually specified for each function.

To assess source-sink status of existing populations, use the `site.analysis` function. Following the example provided in the help documentation, run the function.

```
site_results <- site.analysis(sites = site.dat[,1:3],
  pop.abun = site.dat[,4:5],
  met.size = site.dat[,6:7],
  prop.philo = 0.95,
  sd.philo = 0.05,
  lower.upper_philo=c(lower=0, upper=1),
  lower.upper_survive = c(lower=0, upper=1),
```

```
dispersal = 25,
sd.dispersal = 10,
lower.upper_dispersal = c(lower=10, upper=Inf),
iterations = 5,
seed = 123)
```

The results of `site.analysis` contain both a summary data frame as well as the raw data from each Monte Carlo iteration. The summary data frame can be exported and sorted based on the desired rank criteria using the `ssmc_summary` function.

```
site_summary <- ssmc_summary(ssmc_results = site_results,
                             sort = "avg_rank")
str(site_summary)
```

```
## 'data.frame': 25 obs. of 14 variables:
## $ site : int 8 19 23 5 18 15 1 17 20 24 ...
## $ x : num 70.83 78.16 90.49 96.15 7.29 ...
## $ y : num 0.437 0.925 0.915 0.572 0.804 ...
## $ immigr : num 0.4 0.8 1.8 0.6 0.8 0.4 1.2 0.8 0.8 1.6 ...
## $ philo : num 34 38 27 40.6 51 54.4 0 8.8 0 3.6 ...
## $ emig : num 4 3.6 1.6 1.4 3.6 4.2 0 0.6 0 0 ...
## $ pct_immig: num 0.0133 0.0221 0.0632 0.0134 0.017 ...
## $ pct_sink : num 0 0 0 0 0 0 1 0 1 0.2 ...
## $ pct_src : num 1 1 1 1 1 1 0 1 0 0.8 ...
## $ cont_lam : num 0.1986 0.0851 0.0108 0.0253 0.0368 ...
## $ rank_cont: num 2 3 7 6 4 8 1 9 18.5 18.5 ...
## $ delta_lam: num -3.45 -3.28 -4.61 -3.43 -2.7 ...
## $ rank_lamb: num 2 4 1 3 5 7 19 14 6 8 ...
## $ avg_rank : int 1 2 3 4 4 6 7 8 9 10 ...
```

It is possible to sort the summary data frame using one of the three rank variables generated during the simulation:

1. 'rank_cont': Populations are ranked based on their average contribution, λ_i , to metapopulation capacity
2. 'rank_lamb': Populations are ranked based on the average change in metapopulation capacity, $\Delta\lambda_M$
3. 'avg_rank': The average of 'rank_cont' and 'rank_lamb'

Elements of the summary data frame include:

1. `immig`: The average number of immigrants entering each population
2. `philo`: The average number of philopatric individuals
3. `emig`: The average number of emigrants dispersing from a population
4. `pct_immig`: Average percentage of a population that is comprised of immigrants
5. `pct_sink`: The frequency that a population acted as a sink (i.e. $\frac{i}{p} > 1$) across all simulations
6. `pct_src`: The frequency that a population acted as a source (i.e. $\frac{i}{p} \leq 1$) across all simulations
7. `cont_lamb`: The contribution of each location to the metapopulation (λ_i)
8. `rank_cont`: The rank order importance of each location to the metapopulation
9. `delta_lam`: The change in the value of the the leading eigen value of the connectivity matrix when population i is removed, which is interpreted as the metapopulation capacity. This is calculated as the log ratio of values $\Delta\lambda_M = \log(\lambda_{new}/\lambda_{old})$

10. `rank_lamb`: The rank order importance of each location to metapopulation capacity;
11. `avg_rank`: The rank importance of each potential location, determined by averaging `rank_cont` and `rank_lamb`

Comments on using `site.analysis`

1. This is a rather computationally intensive process, so the number of Monte Carlo iterations has been kept to a minimum for example purposes. It is highly recommended that 1,000–10,000 iterations be used when conducting real analyses.
2. Because `met.size` was specified, there is no need to specify `prop.survive` and `sd.survive`.
3. `met.size`, if provided, must be in standard units with a mean of zero and standard deviation of one
4. The units for dispersal distance are arbitrary, but must correspond with the distances that will be calculated between populations based on the xy coordinates. The model was originally applied to populations with xy coordinates in UTM, and dispersal distances in meters.
5. The `lower.upper_` parameters are used to specify lower and upper threshold values when sampling from a normal distribution.

Assessing the potential of a new or restored population to contribute to the metapopulation

The use of the `best.locale` function is identical to `site.analysis` except that a data frame of potential sites is provided to be evaluated as part of the metapopulation network. `potential.dat` is provided with the package as an example of a potential location data frame.

```
best_results <- best.locale(sites = site.dat[,1:3],
                           potential.sites = potential.dat,
                           pop.abun = site.dat[,4:5],
                           met.size = site.dat[,6:7],
                           prop.philo = 0.95,
                           sd.philo = 0.05,
                           lower.upper_philo=c(lower=0, upper=1),
                           lower.upper_survive = c(lower=0, upper=1),
                           dispersal = 25,
                           sd.dispersal = 10,
                           lower.upper_dispersal = c(lower=10, upper=Inf),
                           iterations = 5,
                           seed = 123)
```

At each Monte Carlo iteration, the function evaluates the contribution of each potential location in contributing to λ_i as well as the metapopulation capacity, λ_M , of the network with the population included. Because these are hypothetical populations with no abundance, size, or survival parameters, these parameters are set to the mean value of all non-zero populations for that iteration of the analysis. As such, each potential location is evaluated as if it were an ‘average’ population on the landscape.

The output from the `best.locale` analysis is similar to that of `site.analysis` and the summary data frame can again be extracted and sorted using `ssmc_summary`

```
best_summary <- ssmc_summary(ssmc_results = best_results,
                             sort = "avg_rank")
str(best_summary)
```

```
## 'data.frame':   25 obs. of  8 variables:
## $ site        : int  6 1 12 16 2 10 14 15 17 18 ...
## $ x           : num  99.42 25.45 6.64 34.05 62.17 ...
```

```
## $ y      : num  6.05 75.19 55.1 74.65 53.28 ...
## $ cont_lam : num  0.0207 0.2007 0.2007 0.2007 0.2007 ...
## $ rank_cont: num  9 2.5 2.5 2.5 2.5 14 14 14 14 14 ...
## $ lambda   : num  2.12 1.88 1.88 1.88 1.5 ...
## $ rank_lamb: num  3 13 13 13 17 6.5 6.5 6.5 6.5 6.5 ...
## $ avg_rank : int  1 2 2 2 5 6 6 6 6 6 ...
```

Citations

Altwegg, R., and H.-U. Reyer. 2003. Patterns of natural selection on size at metamorphosis in water frogs. *Evolution* 57:872–882.

Hanski, I., and O. Ovaskainen. 2000. The metapopulation capacity of a fragmented landscape. *Nature* 404:755–758.