

```
> library(knitr)
> opts_chunk$set(
+ concordance=TRUE
+ )
```

Landscape Occupation Simulation in Dynamic Landscapes

Mestre, F.; Canovas, F.; Pita, R.; Mira, A.; Beja, P.

December 20, 2017

1 Introduction

This vignette explains the process of landscape and metapopulation simulation, using the package `MetaLandSim`. Previously, a crucial step has to be carried out, the parametrization of the SPOM model, using the function `parameter.estimate`. A vignette is available, detailing the process.

2 Landscape Simulation

Landscape simulation and dynamics are carried out respectively by the functions `rland.graph` and `span.graph`. Both these functions allow a different set of random landscapes (represented as graphs) to be created and to suffer dynamic changes, such as patch loss and habitat loss. In `MetaLandSim`, a landscape is represented as:

```
> library(MetaLandSim)
> rl <- rland.graph(mapsize = 1000, dist_m = 60,
+                  areaM = 0.5, areaSD = 0.2, Npatch = 70,
+                  disp = 100, plotG = TRUE)
```

3 Occupation Simulation

The transition from an occupied landscape in time step t to time step $t+1$ can be achieved using the `spom` function, as described by Hanski (1999). This function runs within `simulate.graph` which sequentially simulates the occupation in a landscape throughout several time steps. The occupation and transition between time steps is carried out, respectively, by the functions `'species.graph'` and `'spom'`, as follows:

```
> library(MetaLandSim)
> #The occupation of a landscape is simulated by:
> sp_t0 <- species.graph(rl=rl, method="percentage", parm=50,
```

```

+                               nsew="none", plotG=TRUE)
> names(sp_t0)

[1] "mapsize"                "minimum.distance"    "mean.area"
[4] "SD.area"                "number.patches"      "dispersal"
[7] "distance.to.neighbours" "nodes.characteristics"

```

After loading the species parameters (param1), the transition is made by:

```

> data(param1)
> sp_t1 <- spom(
+ sp_t0,
+ kern="op1",
+ conn="op1",
+ colnz="op1",
+ ext="op1",
+ param_df=param1,
+ betal=NULL,
+ b=1,
+ c1=NULL,
+ c2=NULL,
+ z=NULL,
+ R=NULL
+ )
> #Which has the following elements:
> names(sp_t1)

[1] "mapsize"                "minimum.distance"    "mean.area"
[4] "SD.area"                "number.patches"      "dispersal"
[7] "distance.to.neighbours" "nodes.characteristics" "turnover"

```

4 Sample Work-flow

Although this procedure can be carried out using the functions mentioned above, it is easier to complete the full simulation using only one function that runs all the others internally, while allowing for a repetition of the process, the function is 'iterate.graph'.

Here the simulation process will run only with 2 iterations, for demonstration:

```

> #Loading species parameters
>
> data(param1)
> #Simulating occupation in dynamic landscape
>
> it1 <- iterate.graph(

```

```

+ iter = 2,
+ mapsize = 1000,
+ dist_m = 30,
+ areaM = 0.5,
+ areaSD= 0.1,
+ Npatch = 200,
+ disp = 800,
+ span = 100,
+ par1 = "stoc",
+ par2 = 2,
+ par3 = 2,
+ method = "percentage",
+ parm = 50,
+ nsew = "none",
+ succ = "none",
+ param_df = param1,
+ kern = "op1",
+ conn = "op1",
+ colnz = "op1",
+ ext = "op1",
+ b = 1,
+ graph = FALSE
+ )

Completed iteration 1 of 2
Completed iteration 2 of 2

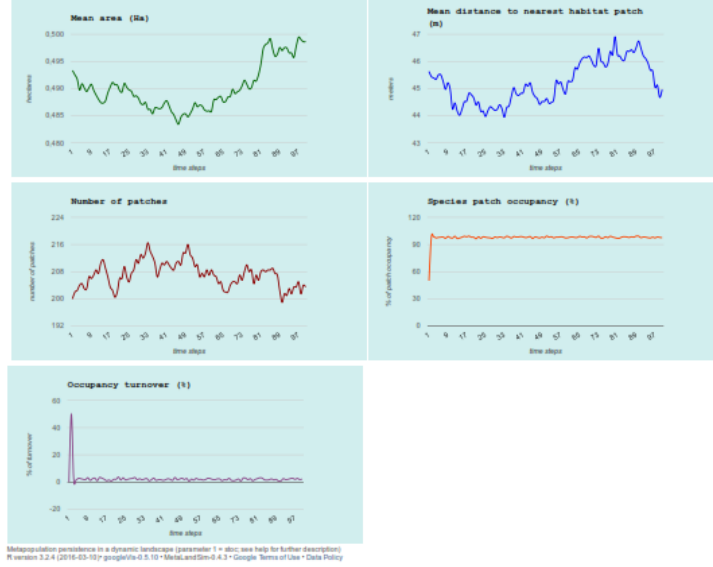
> #This file is composed by the following elements:
>
> names(it1)

[1] "mean_area"      "mean_distance"  "number_patches" "occupancy"
[5] "turnover"

```

However, when running this function the argument 'graph' must be set to TRUE in order to produce graphs with the evolution of the simulations.

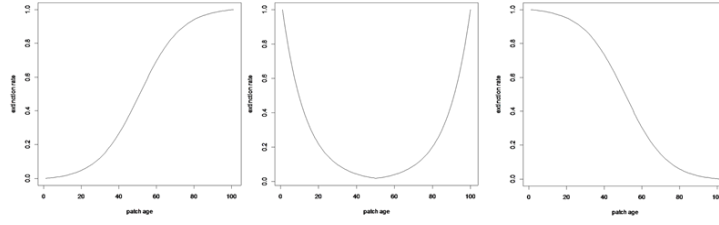
This simulation produces the following graphic output (which opens in an html page):



As a result the user will have a large number of simulations which represent the occupation of a species with a given set of characteristics (as defined by the parameters) in a dynamic landscape. The advantage of this approach is that it requires less parameters which can be estimated from real occupancy or turnover data. It does not require demographic data, the parameters can be derived using only patch occupancy data of one snapshot or sampling session (ideally more).

Here the simulation procedure is repeated only twice (parameter 'iter'), although more simulations have to be run in order to obtain robust results. However, depending on computing power, this simulation can be highly time-consuming (from hours to several days).

After version 0.5 of MetaLandSim an additional option was made available to the users: the argument 'succ'. This allows to chose different options regarding the species preference relating the successional stage of habitat patches, with the following options: 'none' - No discrimination regarding patch successional stage; 'early' - The species prefers patches in an earlier successional stage; 'mid' - The species prefers patches in the mid of the succession; 'late' - The species prefers patches in an later successional stage. This new option includes an additional factor to the extinction probability, changing it with patch age, as follows:



In the above figure it is visible that, when succ='early', the extinction concerning successional stage is lower for younger patches (left); when succ='mid', the extinction concerning successional stage is lower for habitat patches with intermediate ages; when succ='late', the extinction concerning successional stage is lower for older patches. This extinction factor is combined with the one derived from the application of the SPOM model considered (generally depending on patch area).

5 References

1. Hanski I. (1999). Metapopulation Ecology. Oxford University Press. 312 pp.