The cellulaR Package: User Manual

DuccioRocchini

November 4, 2025

Overview

cellulaR is an R package designed to simulate spatial ecological dynamics using cellular automata (CA). The package supports three main modelling functions: a neutral spread model (c.neutral()), a fractal-landscape generator (c.fractal()), and a weighted spread model over heterogeneous landscapes (c.weighted()). These can be used sequentially or independently to explore how spatial structure and local interactions drive vegetation or species spread.

Installation

You can install the package from GitHub using:

```
install.packages("remotes")
remotes::install_github("docciorocchini/cellulaR")
library(cellulaR)
```

1 Function Reference

c.neutral()

Description: Simulates vegetation or species spread over a homogeneous grid without spatial heterogeneity. Growth (colonisation) and mortality are governed by neighbourhood interactions and uniform probabilities.

```
c.neutral(num_iterations = ,
          grid_size = ,
          colonisation_rate = ,
          mortality_rate = ,
          plot_interval = )
```

Arguments:

- `num_iterations` – integer; number of time steps.
- `grid_size` – integer or vector; dimension of the grid (e.g., 50 for 50×50).
- `colonisation_rate` – numeric; baseline probability of spread.
- `mortality_rate` – numeric; baseline probability of die-off.
- `plot_interval` – integer; how often to generate plots.

Value: Returns a list containing (a) the time-series of grid states at each iteration (or at the plotting interval), (b) summary statistics of cover and patch dynamics, (c) the parameter list used.

```
library(cellulaR)
sim1 <- c.neutral(num_iterations = 100,
                  grid_size = 50,
                  colonisation_rate = 0.04,
                  mortality_rate = 0.01,
                  plot_interval = 10)
```

```
c.fractal()
```

Description: Generates a synthetic fractal landscape (via Perlin or similar noise) to represent spatial heterogeneity in environmental suitability or resources. This landscape can then serve as input for CA models of spread.

```
c.fractal(n = ,
          octaves = ,
          frequency = ,
          amplitude = )
```

Arguments:

- `n` – integer; side length of the square grid (e.g., 100 for 100×100).
- `octaves` – integer; number of noise layers.
- `frequency` – numeric; base frequency of the noise.
- `amplitude` – numeric; amplitude of the noise relative to base value.

Value: Returns a matrix (or raster) of dimension $n \times n$ with continuous values representing suitability or resource availability.

```
land <- c.fractal(n = 100,
                  octaves = 4,
                  frequency = 0.1,
                  amplitude = 1.0)
```

`c.weighted()`

Description: Combines the spread model of `c.neutral()` with the heterogeneous landscape generated by `c.fractal()`. Colonisation, mortality and spread probabilities are modulated by a weight field (suitability map) and neighbourhood influences, producing more ecologically realistic spatial dynamics.

```
c.weighted(num_iterations = ,
           landscape = ,
           weight_field = ,
           colonisation_rate = ,
           mortality_rate = ,
           plot_interval = )
```

Arguments:

- `num_iterations` – integer; number of time steps.
- `landscape` – matrix or raster; the base grid of suitability values (from `c.fractal()`).

- `weight_field` – matrix or raster; same or derived from `landscape`, used to modulate transition probabilities.
- `colonisation_rate` – numeric; baseline colonisation probability.
- `mortality_rate` – numeric; baseline mortality probability.
- `plot_interval` – integer; the plotting interval.

Value: Returns an object (e.g., of class `cellulaR_sim`) containing:

- **Landscape state time-series:** A list or array of grid states over iterations, enabling visualisation of spread dynamics.
- **Summary statistics:** Data.frame or list of metrics such as vegetated area over time, colonisation events, mortality counts, patch sizes.
- **Parameter and weight field data:** The input parameters used and the weight map used for the simulation, enabling reproducibility and downstream analysis.

```
weighted_sim <- c.weighted(num_iterations = 200,
                           landscape = land,
                           weight_field = land,
                           colonisation_rate = 0.05,
                           mortality_rate = 0.02,
                           plot_interval = 20)
```

Further Notes

The three functions are designed to work sequentially: generate a neutral spread baseline (`c.neutral()`), create a heterogeneous suitability surface (`c.fractal()`), and then run a spread simulation over that surface (`c.weighted()`). Users can of course use any combination. Typical workflows include exploring how changes in weight field structure or transition rates affect emergent spatial patterns, fragmentation and connectivity of vegetation or species patches.

References

Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature.* W.H.Freeman Company.