Peer Community Journal

Section: Ecology

Research article

Published
2024-10-08

Cite as
Alexandre Génin, Guillaume Dupont, Daniel Valencia, Mauro Zucconi, M. Isidora Ávila-Thieme, Sergio A. Navarrete and Evie A. Wieters (2024) *Easy, fast and reproducible Stochastic Cellular Automata with chouca*, Peer Community Journal, 4: e95.

Correspondence
alexandre.genin@sete.cnrs.fr

Peer-review
Peer reviewed and recommended by PCI Ecology,
https://doi.org/10.24072/pci.ecology.100686

Easy, fast and reproducible Stochastic Cellular Automata with chouca

Alexandre Génin[1,2,3], Guillaume Dupont[2,4], Daniel Valencia[2], Mauro Zucconi[2,5], M. Isidora Ávila-Thieme[6,7,8,9], Sergio A. Navarrete[2,5,10,8], and Evie A. Wieters[2]

Volume **4** (2024), article e95

https://doi.org/10.24072/pcjournal.466

Abstract

Stochastic cellular automata (SCA) are models that describe spatial dynamics using a grid of cells that switch between discrete states over time. They are widely used to understand how small-scale processes scale up to affect ecological dynamics at larger spatial scales, and have been applied to a wide diversity of theoretical and applied problems in all systems, such as arid ecosystems, coral reefs, forests, bacteria, or urban growth. Despite their wide applications, SCA implementations are often ad-hoc, lacking performance, guarantees of correctness and poorly reproducible. *De novo* implementation of SCA for each specific system and application also represents a major barrier for many practitioners. To provide a unifying, well-tested technical basis to this class of models and facilitate their implementation, we built chouca, an R package that translates definitions of SCA models into compiled code, and runs simulations in an efficient way. chouca supports SCA based on rectangular grids where transition probabilities are defined for each cell, with performance typically two to three orders of magnitude above typical implementations in interpreted languages (e.g. R, Python), all while maintaining an intuitive interface in the R environment. Exact and mean-field simulations can be run, and both numerical and graphical results can be easily exported. Besides providing better reproducibility and accessibility, a fast engine for SCA unlocks novel, computationally intensive statistical approaches, such as simulation-based inference of ecological interactions from field data, which represents by itself an important avenue for research. By providing an easy and efficient entry point to SCAs, chouca lowers the bar to the use of this class of models for ecologists, managers and general practitioners, providing a leveled-off reproducible platform while opening novel methodological approaches.

[1] Copernicus Institute of Sustainable Development, Utrecht University – Utrecht, The Netherlands, [2] Estación Costera de Investigaciones Marinas (ECIM) and Núcleo Milenio para la Ecología y Conservación de los Ecosistemas de Arrecifes Mesofóticos Templados (NUTME), Pontificia Universidad Católica de Chile – Las Cruces, Chile, [3] Experimental and Theoretical Ecology Station, Centre National de la Recherche Scientifique – Moulis, France, [4] International Master of Science in Marine Biological Resources (IMBRSea), Ghent University – Ghent, Belgium, [5] Centro de Investigación Oceanográfica en el Pacifico Sur-Oriental (COPAS COASTAL), Universidad de Concepción – Concepción, Chile, [6] Escuela de Negocios, Facultad de Ciencias Sociales y Artes, Universidad Mayor – Temuco, Chile, [7] Center for Resilience, Adaptation and Mitigation (CReAM), Universidad Mayor – Temuco, Chile, [8] Instituto Milenio en Socio-Ecología Costera (SECOS), Pontificia Universidad Católica de Chile – Santiago, Chile, [9] Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile – Santiago, Chile, [10] Center for Applied Ecology and Sustainability (CAPES), Pontificia Universidad Católica de Chile – Santiago, Chile

CENTRE MERSENNE

Peer Community Journal is a member of the Centre Mersenne for Open Scientific Publishing
http://www.centre-mersenne.org/

e-ISSN 2804-3871

Introduction

The analysis of spatial patterns has proven essential to understand ecological system dynamics, and various modelling approaches have helped ground empirical patterns into ecological theory. Among such approaches, models based on Stochastic Cellular Automata (hereafter SCA), also called Probabilistic or Random Cellular Automata, or Locally-interacting Markov Chains, have been a particularly useful, heuristic and widely used approach (Wolfram, 1984; Louis & Nardi, 2018). Cellular automata are based on a grid of cells that switch over time between a finite number of states. Most often, SCA are considered over a rectangular grid, though other geometries can exist (van Baalen, M., 2000). A famous deterministic cellular automaton (CA) is Conway's game of life, which is defined by two discrete states ("dead" and "alive") and a set of deterministic rules to make cells switch between them (Gardner, 1970; Bays, 2010). Stochastic cellular automata follow the same principles, but state transitions occur with a given probability instead of being based on deterministic rules. The probabilities of a cell switching from one state to another is assumed to depend on model parameters, the global state of the system (the proportion of cells in each state), and the local neighborhood of the focal cell. In all cases, the system future dynamics is probabilistically defined by its current state, *i.e.* dynamics are memoryless.

The use of SCA in ecology is widespread, as they have been used to describe the dynamics of a large array of ecosystems, including mussel beds (Guichard et al., 2003), arid ecosystems (Kéfi et al., 2007), forests (Heinonen & Pukkala, 2007), rocky shores (Wootton, 2001), coral reefs (Muthukrishnan et al., 2016; Génin et al., 2024) and plant communities (Lanzer & Pillar, 2002). SCA are often used to understand how local processes can scale up to affect landscape-wide properties, such as the persistence or extinction of a given species, the type of spatial patterns arising at the scale of a landscape (Pascual et al., 2002), or the spread of fire or epidemics. The latter case is a classical application of SCA in applied ecology, where data on local processes affecting forest stands can be coupled with GIS data to provide guidance on forest fire sensitivity (Yassemi et al., 2008). Though common in ecosystem modelling, and the primary purpose for this package, SCA are useful beyond this sole discipline, and are for example also used to describe epidemics (Yakowitz et al., 1990; Keeling, 2000), tumor growth (Moreira & Deutsch, 2002) or organism development (Manukyan et al., 2017).

The popularity of SCA is probably rooted in their relatively light need for formal mathematics compared to other approaches modelling spatial processes (*e.g.* partial derivative equations): only the probabilities of transitions between states need to be defined. The drawback of this simplicity is that the numerical simulation of SCA is typically computationally-intensive. Current approaches to do so efficiently rely on approximations, either assuming no spatial structure (mean field approximation), or using approximations to obtain analytical results, such as spatial moment equations (Lion, 2016; of which pair approximation is a specific case; Matsuda et al., 1992; Iwasa, 2000). However, these approximations are inappropriate when long-range correlations occur within a landscape (Iwasa, 2000), or when the full simulated landscape is needed as a model output, for example to compute spatial metrics (Génin et al., 2018). In many cases, the explicit numerical simulation must be run, which is often done on grids that may not be large enough to fully capture emergent spatial patterns (van de Koppel et al., 2011; Majumder et al., 2021). On top of these performance issues, most ecological studies based on SCA come with their own implementation. This opens the possibility of errors in code and often makes it difficult to reproduce model simulations. We aim at alleviating these issues with chouca, which provides a unifying and well-tested technical basis to SCA. Our goal is to improve the performance and accessibility of this class of models, and ultimately allow ecologists to spend more time exploring the behavior of their models, rather than on their implementation.

Supported models

The R package chouca works with 2-dimensional rectangular grids of cells (a “landscape”). Each cell can be in one of a finite set S of n elementary states $S_1 \dots S_n$. Probabilities of transition are assumed to depend only on (i) the proportion of neighbors in each state, captured by the vector $\boldsymbol{q} = (q_1, \dots, q_n)$, (ii) the proportion of cells in a given state in the whole landscape, $\boldsymbol{p} = (p_1, \dots, p_n)$, and (iii) a set of constant model parameters $\boldsymbol{\theta}$.

chouca has been primarily designed for modeling the dynamics of sessile organisms over space, which do not move and reproduce through the dispersal of propagules (*e.g.* gap-models; Shugart & West, 1981). This includes a wide range of organisms, ranging from forests, herbaceous plants or coral reefs, but the supported models may also be useful to describe other ecological situations. For the moment, the types of models that can be implemented exclude cellular automata in which an intermediate distance of interaction is considered (*e.g.* through a dispersion kernel; Muthukrishnan et al., 2016), or those in which a preferential direction exists (*e.g.* modeling water redistribution on a slope; Mayor et al., 2013), though these limitations are planned to be lifted in later versions. More importantly, SCA that are based on the dynamics of pairs of cells, for instance in which two cells swap their respective state, cannot be implemented directly. This is typical when representing the movements of an organism in a landscape (Pascual et al., 2002), and we provide links to alternative software at the end of the article for these models.

In the package, the probabilities of transition of a cell from state i to j, $P(S_i \to S_j)$, can be any function of $\boldsymbol{p}$, $\boldsymbol{q}$, and $\boldsymbol{\theta}$ – however, it will work best and without approximation if it has the following form:

$$(1) \qquad P(S_i \to S_j) = a_0 + g_1(q_1) + \cdots + g_n(q_n) + \zeta(\boldsymbol{q}, \boldsymbol{q}) + \zeta(\boldsymbol{q}, \boldsymbol{p}) + \zeta(\boldsymbol{p}, \boldsymbol{p})$$

where, for any transition $S_i \to S_j$, a_0 is a constant, g_s is any univariate function of q_s, and $\zeta(\boldsymbol{x}, \boldsymbol{y})$ is the sum, defined for two vectors $\boldsymbol{x} = (x_1, \dots, x_n)$ and $\boldsymbol{y} = (y_1, \dots, y_n)$ as

$$(2) \qquad \zeta(\boldsymbol{x}, \boldsymbol{y}) = a_1 x_1^{\alpha_1} y_1^{\beta_1} + a_2 x_1^{\alpha_2} y_2^{\beta_2} + a_3 x_2^{\alpha_3} y_1^{\beta_3} + \cdots + a_K x_n^{\alpha_K} y_n^{\beta_K}$$

in which the $(a_i)_{i \in [1:K]}$, $(\alpha_i)_{i \in [1:K]}$ and $(\beta_i)_{i \in [1:K]}$ are constants and K is the total number of terms of the sum. In practice, this functional form is flexible enough to approximate the probabilities of transition of many ecological models.

Implementing and working with an SCA in chouca typically consists in four steps, in which the user (1) defines the model states and the transitions between them, (2) creates an initial landscape (grid of cells), (3) runs the model, and (4) displays or extracts the results (Figure 1). We detail in this paper this workflow (Figure 1) – documentation is available throughout the package, individually for each function, or as a whole in an R “vignette”, accessible with the command `vignette(“chouca-package”)`.

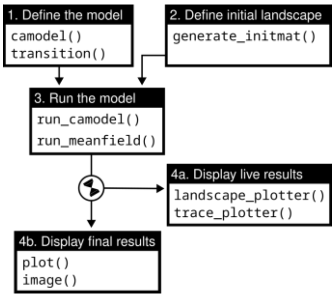

Figure 1 - Main tasks (boxes) of the chouca package, and their associated sets of functions

Example applications

A simple model of mussel bed

To illustrate how a stochastic cellular automaton can be defined with chouca, we use the model of Guichard et al. (2003), which describes the dynamics of mussels colonizing rocks exposed to waves. This model has three cell states (i) "disturbed", (ii) "empty" and (iii) "occupied" (by mussels). During a single time-step, a disturbed cell becomes an empty cell with probability 1. Such transition can be defined by using a call to the R function `transition()`:

```
transition(from = "disturbed", to = "empty", ~ 1)
```

This statement declares a transition from a "disturbed" state to an "empty" state, with the last argument being a symbolic expression starting with "~", that describes how to compute the probability, here being simply equal to the constant "1".

The model assumes that the establishment of new individuals always occurs next to existing mussels. In other words, for a focal cell in the "bare" state, its probability of switching to the "mussel" state is not constant, but given by αq_{mussel}, where α is a productivity rate, and q_{mussel} is the proportion of cell neighbors in the mussel state. Such transition is defined by the following call in R:

```
transition(from = "empty", to = "mussel", ~ alpha * q["mussel"])
```

Here, `q["mussel"]` is used to refer to the proportion of cell neighbors in the "mussel" state, a continuous number between 0 and 1. Similarly, `p["mussel"]` can be used to refer to the global proportion of cells in the landscape in the "mussel" state.

Mussels can be perturbed by incoming waves, which dislodge them and turn them into "disturbed" cells. In this model, the probability of a mussel cell to become disturbed is the sum of a baseline term δ, and an additional term d, which is non-zero only if the mussel cell has one or more disturbed neighbors:

```
transition(from = "mussel", to = "disturbed", ~ delta + d * ( q["disturbed"] > 0 ))
```

The original model considers that cells are neighbors when they share an edge (current options include a 4-way or von-Neumann neighborhood, the other option being a Moore or 8-way neighborhood), and uses a toric space for simulations, meaning that the up/leftmost cells of the grid are neighbors of the bottom/rightmost cells. Putting everything together, this model can be defined using the following syntax:

```
musselbed_mod <- camodel(
  transition(from = "disturbed", to = "empty",  ~ 1),
  transition(from = "empty", to = "mussel",     ~ alpha * q["mussel"]),
  transition(from = "mussel", to = "disturbed", ~ delta + d * ( q["disturbed"] > 0 )),
  parms = list(alpha = 1, delta = 0.2, d = 1), # parameters
  wrap = TRUE, # toric space
  neighbors = 4 # 4-way neighborhood
)
```

The call to the `camodel()` function above estimates the constants in the functional form described in equation 1 to match the parameters of the model. If this process yields large residual error, for example because the transition probabilities do not correspond to the functional form in equation 1, a warning is produced. The structure of the model can be displayed on the R console using `print()`, or as a graph using the generic function `plot()` (Figure 2).

```
> print(musselbed_mod)
Stochastic Cellular Automaton
```

```
States: disturbed empty mussel

Transition: disturbed -> empty
  ~   1
Transition: empty -> mussel
  ~   alpha * q["mussel"]
Transition: mussel -> disturbed
  ~   delta + d * (q["disturbed"] > 0)

Neighborhood: 4x4
Wrap: TRUE
Max error: 0 (OK)
Max rel error: 0 (OK)
```

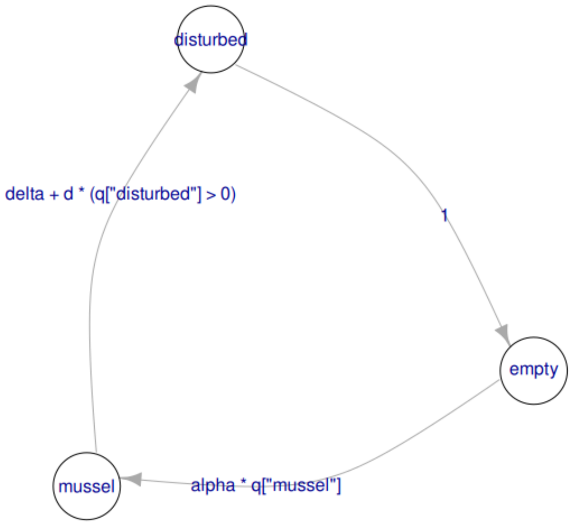

Figure 2 - Structure of the mussel bed model displayed as a graph using `plot(musselbed_mod)`, in which nodes are states, and arrows represent the possible transitions, with the expression used to compute their probabilities

Once the model is created, an initial landscape can be defined with a given number of rows and columns using `generate_initmat()`, which creates a landscape with randomly-distributed cell states in space, but following the specified initial covers:

```
init_landscape <- generate_initmat(musselbed_mod,
                                   pvec = c(disturbed = 0.1, empty = 0.1, mussel = 0.8),
                                   nrow = 64, ncol = 64)
```

The model can then be simulated for a given number of time steps, below from 0 to 50, using `run_camodel()`. Standard methods such as `plot()` or `image()` can be used to display the global covers after the model has run, or the resulting landscapes:

```
output <- run_camodel(musselbed_mod, init_landscape, times = seq(0, 50))
plot(output)
image(output)
```

By default, chouca uses a C++ backend based on Rcpp (Eddelbuettel, 2013), which has reasonable performance. This can be improved by compiling the model code just before the simulation is run, and using memoization so that transition probabilities are computed only once for cells with the same neighborhood configuration. This typically improve performance by two orders of magnitude over a typical implementation (Figure 3; Schneider et al., 2016), which can be further increased by parallelizing computations over multiple cores, though this is a less efficient approach. Enabling these options can be done by passing control arguments as an R list object:

```
control_args <- list(engine = "compiled",
                     cores = 4,
                     precompute_probas = TRUE)
output <- run_camodel(musselbed_mod, init_landscape, niter = 256, control = control_args)
```

This "control" list defines how the simulation is run, which data to save from the simulation, or the textual output to print while the simulation is running (a complete list of options can be found in the help page by using `?run_camodel` on the R console). The user can also supply custom functions that will be run as the simulation is running. This can be useful, for example, to display landscapes, covers, or compute statistics on the 2D landscape as the simulation is running, which we illustrate in the following sections.

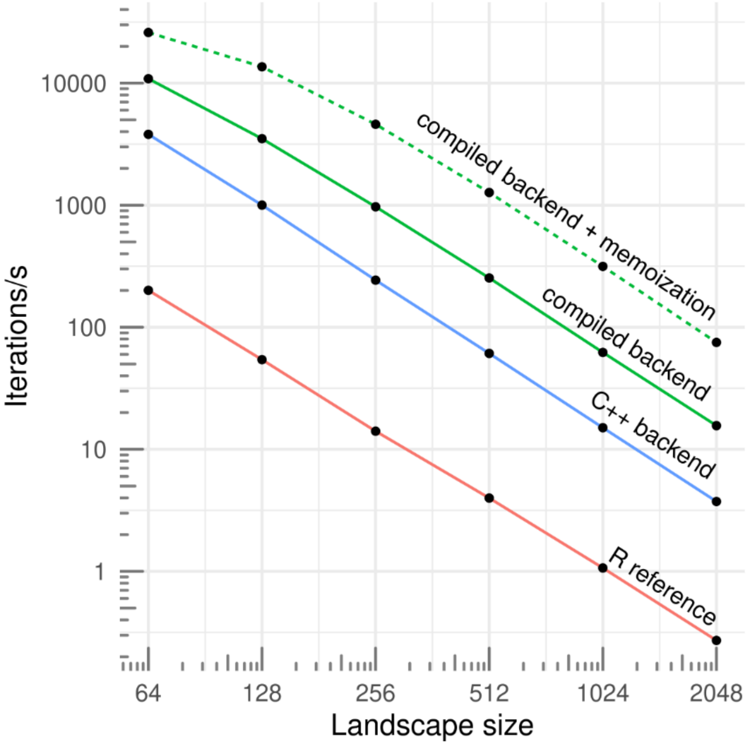

Figure 3 - Simulation speeds for a set of three simple ecological models (2-3 states and 3-4 transitions) according the the grid size, for a pure-R implementation (R reference; Schneider et al., 2016), and three backends provided by chouca (blue and green lines). Single-core performance on a 2020 desktop computer.

Graphical explorations

Because an SCA describes dynamics over landscapes, they are particularly well-adapted to pattern-oriented modelling (Grimm et al., 1996), in which a model is defined and revised based on a qualitative or quantitative comparison with empirical patterns. Likelihood-based approaches are increasingly popular to compare models with data (Hartig et al., 2011 and section below), but the qualitative comparison and visual exploration of model dynamics remains an essential phase of the design of spatial models. To make this modeling step more accessible, we made it easy to investigate visually the behavior of models, and illustrate here this approach with an epidemiological example.

Keeling (2000) uses an SCA-based approach to investigate the spread of a parasite over space, with an application to forests. This very simple model uses three states, “host”, “parasitized”, “empty”, and can be defined as follows in chouca:

```
mod <- camodel(transition(from = "empty", to = "host",
                          ~ 1 - ( 1 - g )^( 4 * q["host"] ) ),
               transition(from = "host", to = "parasitized",
                          ~ 1 - ( 1 - T )^( 4 * q["parasitized"] )),
               transition(from = "parasitized", to = "empty",
                          ~ 1),
               parms = list(g = 0.05, T = 0.5),
               wrap = TRUE,
               neighbors = 4)
```

where g is the growth rate of the host, and T the transmissibility of the parasite (this model assumes that infection is always fatal, so the transition from “parasitized” to “empty” is equal to 1).

This model can produce interesting “epidemiological fronts”, which stem from the way the parasite spreads to its neighboring hosts, killing them and leaving behind empty, bare patches that are later recolonized by the host, albeit at a slower rate. This phenomenon of fronts propagating through an excitable media may be difficult to quantify formally, but can be easily visualized from model outputs. This can be done in chouca by setting the run_camodel() function to display the results as the simulation is run. To do so, we adjust the control list to include a function that displays the landscapes:

```
options <- list(custom_output_fun = landscape_plotter(mod),
                custom_output_every = 1)
```

then run the model on a 256×256 grid seeded with 10% of cells in the “parasitized” state:

```
initmm <- generate_initmat(mod, c(host = 0.9, parasitized = 0.1, empty = 0),
                           nrow = 256, ncol = 256)
output <- run_camodel(mod, initmm, times = seq(0, 1024), control = options)
```

The above lines of code will run the model and display the changing landscape, which allows investigating the spreading patterns of the parasite. Once this visualization step is done (Figure 4, and animated version in SM1), the options set to visualize the landscape can be removed to reduce simulation times, for example to investigate the model behavior along a range of parameter values.

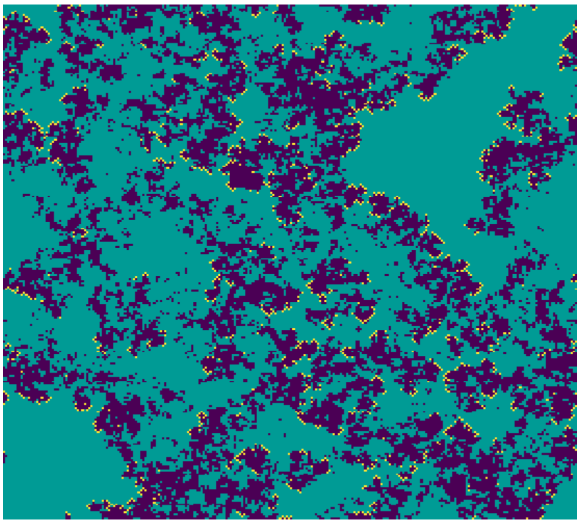

Figure 4 - Landscape patterns of the host-parasitized-empty model (respectively green, yellow and blue) as displayed on screen.

Inference of local interactions from landscape-scale patterns

Because SCA are defined on grids, a natural application is to compare their output to empirical raster data, such as remote-sensing images, to infer local-scale ecological interactions from landscape-wide spatial patterns. Arid systems provide a good illustration of this approach: in those systems, interactions between plants are often a balance of negative effects, through competition for nutrients or water, with positive effects, for example with an increased survival of seedlings below the canopy of taller plants (Valiente-Banuet & Ezcurra, 1991). These effects can be strong enough for new plants to mostly establish below existing plants, which results in their aggregation into patches, and has important consequences for the resilience of those systems to changes in aridity (Kéfi et al., 2007). The sizes and numbers of those patches can be readily quantified from remote-sensing images, and such patterns can be used to infer whether facilitation occurs between plants (Xu et al., 2015; Chen et al., 2022). This is traditionally done by summarizing the spatial structure into spatial statistics, such as spatial autocorrelation (Sankaran et al., 2017) or type of patch size distribution (Siteur et al., 2023; Pichon et al., 2024) and linking the observed changes in those metrics to theoretical results (Scanlon et al., 2007; Kéfi et al., 2024). However, such qualitative comparison logically results in a qualitative and corroborative result, *i.e.* is there or is there not facilitation between plants, rather than a more informative quantitative result, *i.e.* how strong facilitation is between plants. A quantitative inference must be based on an approach that links quantitatively some aspects of the spatial patterns to the strength of facilitation. This can be done by defining a model that links the strength of facilitation with the expected patterns, and finding model parameters that maximize agreement between model output and data. This is expensive computationally, but this limitation is alleviated by a fast SCA engine such as chouca, as we show below.

We define here a model of an arid ecosystem with two states, “bare” and “vegetated”. A bare cell can become a vegetated cell with the probability p_{plant}. A vegetated cell can become bare (plants die) with the probability $d(1-\beta q_{plant})$, where d is a constant mortality rate, that is reduced by the coefficient β. β captures the local effect of plants, either facilitation when they increase the survival of plants near existing vegetation ($\beta > 0$), or competition when survival is decreased ($\beta < 0$). This model is implemented in chouca using:

```
facilitation_mod <- camodel(
  transition(from = "bare", to = "plant", ~ p["plant"] ),
  transition(from = "plant", to = "bare", ~ d * ( 1 - beta * q[“plant”]) ),
  parms = list(beta = 0.5, d = 0.85),
  wrap = TRUE,
  neighbors = 4
)
```

We ran the model till equilibrium on a 1024×1024 grid, to simulate a landscape that would be obtained from empirical data (*e.g.* a remote sensing image) using $(d,\beta) = (0.85, 0.5)$. From this landscape used as observed data, we computed the distribution of pairs, which summarizes all the possible pairs of neighboring cells present in the landscape $\boldsymbol{n_{obs}} = (n_{p,0}, n_{p,p}, n_{0,0})$. We then define the likelihood $P(\boldsymbol{n_{obs}} \mid d, \beta)$ assuming these observed number of pairs follow a multinomial distribution of size N_p (the sum of all pairs, a fixed number given the grid size and neighborhood type), and probabilities $\boldsymbol{\mu} = (\mu_{p,0}, \mu_{p,p}, \mu_{0,0})$. $\boldsymbol{\mu}$ defines the relative probabilities of observing each type of pair in the grid, which depend on the particular values of d and β, and can be estimated by simulating the landscape with these parameter values. This estimate of the likelihood formally quantifies the agreement between model and data, and allows exploring the parameter space in terms of d and β to find the most likely parameter combinations given the observed patterns.

We find that this approach can recover the parameter values used for d and β, with only one global maximum in the likelihood function (Figure 5; code provided in SM2), showing that the distribution of pairs is a sufficient statistic to recover the model parameters. Applying such an approach to empirical data would require further testing of model assumptions, for example, to investigate whether facilitation occurs on the recruitment of new plants instead of on the mortality of adult plants – this can be done by simply changing the model definition above. Because this approach is likelihood-based, model support can be compared using traditional statistics, such as AIC, and a Bayesian approach can be used to estimate credible intervals on parameter values, or use informative priors grounded in knowledge about the system (Hartig et al., 2011).

Using this type of simulation-based inference is very expensive computationally, as this simple test requires running around a thousand simulations. This would require long-running computations with usual SCA implementations, but takes just under five minutes with chouca on a 2018 laptop computer (8 cores). This way of calibrating models has seldom been used in spatial ecology – a fast SCA implementation is essential to make it more accessible and test its relevance to real-world datasets.

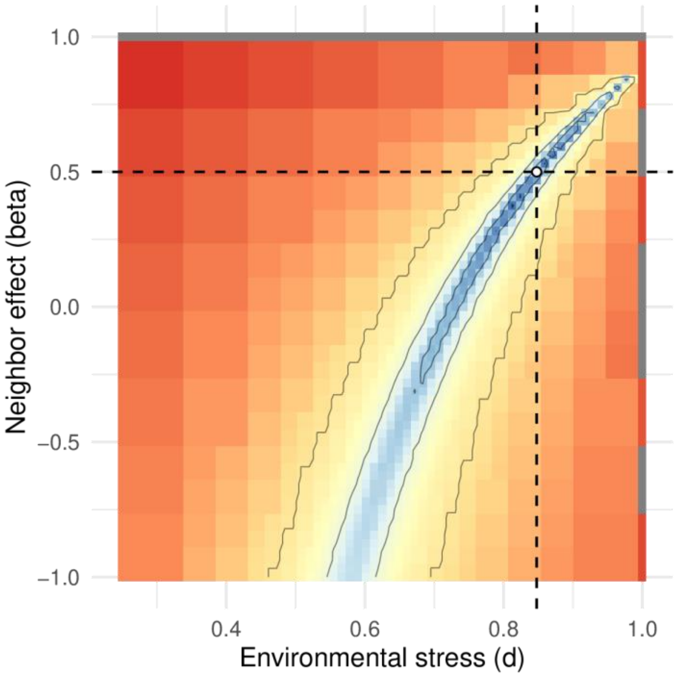

Figure 5 - Likelihood surface as a function of the two model parameters d and β. Blue values indicate parameter combinations with high likelihood. The white dot and lines indicate the estimated parameter values.

Conclusion

chouca is an easy-to-use package to model, simulate, and visualize SCA in a reproducible way, that enables an interactive design and revision of models as well as novel methodological approaches. chouca does not support all types of cellular automata, and does not replace more generic modelling frameworks such as NetLogo (Wilensky, 1999) or simecol (Petzoldt & Rinke, 2007), but allows very efficient simulation for the types of SCA it supports. The package focuses on processes defined at the level of a cell, while other SCA define processes at the level of pair of cells, a model representation other software packages can provide (*e.g.* CellLab-CTS; Tucker et al., 2016). Another limitation is the use of rectangular grids, which may produce discretisation artefacts that are not present in real-world patterns – future improvements may include the ability to run simulations on triangular or hexagonal grids. It is important to note that because chouca splits the definition of an SCA model from its simulation phase, it may use different backends for simulation. This opens the possibility of future improvements to already-implemented models without modification of existing code.

Because of the relatively low bar of entry of SCA compared to other forms of spatial modelling, we hope this work will contribute to make more accessible the testing of hypotheses linking spatial processes to patterns in ecology, as well exploring system-level consequences of specific local management decisions. chouca is available and maintained on CRAN, and welcomes comments, feedback and bug reports on its home page at https://github.com/alexgenin/chouca.

Acknowledgements

This manuscript has received very constructive comments and feedback from Samuel Alizon, Broder Breckling and one anonymous reviewer, to whom we are thankful. Preprint version 6 of this article has been peer-reviewed and recommended by Peer Community In Ecology (https://doi.org/10.24072/pci.ecology.100686; Alizon, 2024).

Data, script and code availability

All code used for this work is freely-available at Zenodo under a CC-BY license (Génin, 2024), available at https://doi.org/10.5281/ZENODO.11567662, along with chouca version v0.1.99 used at the time of writing.

Funding

AG has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement 896159 (INDECOSTAB). MGZ thanks the Pontificia Universidad Católica de Chile for the doctoral student support scholarship, and programs COPAS-COASTAL (FB10021) and Núcleo Milenio NUTME NCN2023_004 for the awarded doctoral thesis fellowships. MIAT acknoweldges support from FONDECYT 3220110, GD from the IsBlue program (ANR-17-EURE-0015), EAW from FONDECYT 1181719, 1241901, and Núcleo Milenio NCN2023_004 (NUTME), and SAN from NUTME ICM_NCN2023_004, SECOS, ICN 2019-015, CAPES, PIA/BASAL FB0002, COPAS COASTAL FB21002, and FONDECYT 1200636.

Conflict of interest disclosure

The authors declare they comply with the PCI rule of having no financial conflicts of interest.

References

Alizon S (2024) An R Package for Flexible and Fast Stochastic Cellular Automata Modeling. *Peer Community in Ecology*, **100686**. https://doi.org/10.24072/pci.ecology.100686

van Baalen, M. (2000) Pair approximations for different spatial geometries. In: The geometry of ecological interactions: simplifying spatial complexity. *Cambridge Studies in Adaptive Dynamics*, pp. 359–387. https://doi.org/10.1017/cbo9780511525537.023

Bays C (2010) Introduction to Cellular Automata and Conway's Game of Life. In: Game of Life Cellular Automata (ed Adamatzky A), pp. 1–7. Springer London, London. https://doi.org/10.1007/978-1-84996-217-9_1

Chen BJW, Teng SN, Zheng G, Cui L, Li S, Staal A, Eitel JUH, Crowther TW, Berdugo M, Mo L, Ma H, Bialic-Murphy L, Zohner CM, Maynard DS, Averill C, Zhang J, He Q, Evers JB, Anten NPR, Yizhaq H, Stavi I, Argaman E, Basson U, Xu Z, Zhang M, Niu K, Liu Q, Xu C (2022) Inferring plant–plant interactions using remote sensing. *Journal of Ecology*, **110**, 2268–2287. https://doi.org/10.1111/1365-2745.13980

Eddelbuettel D (2013) Seamless R and C++ integration with Rcpp. Springer, New York. https://doi.org/10.1007/978-1-4614-6868-4

Gardner M (1970) Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, **223**, 120–123. http://dx.doi.org/10.1038/scientificamerican1070-120

Génin A (2024) Scripts used in the publication “Easy, fast and reproducible Stochastic Cellular Automata with chouca” https://doi.org/10.5281/ZENODO.11567662

Génin A, Majumder S, Sankaran S, Danet A, Guttal V, Schneider FD, Kéfi S (2018) Monitoring ecosystem degradation using spatial data and the R package spatialwarnings. *Methods in Ecology and Evolution*, **9**, 2067–2075. https://doi.org/10.1111/2041-210X.13058

Génin A, Navarrete SA, Garcia-Mayor A, Wieters EA (2024) Emergent spatial patterns can indicate upcoming regime shifts in a realistic model of coral community. *The American Naturalist*, **203**(2), 204-218. https://doi.org/10.1086/728117

Grimm V, Frank K, Jeltsch F, Brandl R, Uchmański J, Wissel C (1996) Pattern-oriented modelling in population ecology. *Science of The Total Environment*, **183**, 151–166. https://doi.org/10.1016/0048-9697(95)04966-5

Guichard F, Halpin PM, Allison GW, Lubchenco J, Menge BA (2003) Mussel disturbance dynamics: signatures of oceanographic forcing from local interactions. *The American Naturalist*, **161**, 889–904. https://doi.org/10.1086/375300

Hartig F, Calabrese JM, Reineking B, Wiegand T, Huth A (2011) Statistical inference for stochastic simulation models - theory and application: Inference for stochastic simulation models. *Ecology Letters*, **14**, 816–827. https://doi.org/10.1111/j.1461-0248.2011.01640.x

Heinonen T, Pukkala T (2007) The use of cellular automaton approach in forest planning. *Canadian Journal of Forest Research*, **37**(11). https://doi.org/10.1139/X07-073

Iwasa Y (2000) Lattice models and pair approximation in ecology. In: The geometry of ecological interactions: simplifying spatial complexity. Cambridge Studies in Adaptive Dynamics, pp. 227–251. Cambridge University Press. https://doi.org/10.1017/cbo9780511525537.016

Keeling MJ (2000) Evolutionary Dynamics in Spatial Host–Parasite Systems. In: The Geometry of Ecological Interactions (eds Dieckmann U, Law R, Metz JAJ), pp. 271–291. Cambridge University Press. https://doi.org/10.1017/CBO9780511525537.018

Kéfi S, Génin A, Garcia-Mayor A, Guirado E, Cabral JS, Berdugo M, Guerber J, Solé R, Maestre FT (2024) Self-organization as a mechanism of resilience in dryland ecosystems. *Proceedings of the National Academy of Sciences*, **121**, e2305153121. https://doi.org/10.1073/pnas.2305153121

Kéfi S, Rietkerk M, Alados CL, Pueyo Y, Papanastasis VP, El Aich A, de Ruiter PC (2007) Spatial vegetation patterns and imminent desertification in Mediterranean arid ecosystems. *Nature*, **449**, 213–217. https://doi.org/10.1038/nature06111

van de Koppel J, Gupta R, Vuik C (2011) Scaling-up spatially-explicit ecological models using graphics processors. *Ecological Modelling*, **222**, 3011–3019. https://doi.org/10.1016/j.ecolmodel.2011.06.004

Lanzer ATS, Pillar VD (2002) Probabilistic cellular automaton: model and application to vegetation dynamics. *Community Ecology*, **3**, 159–167. https://doi.org/10.1556/ComEc.3.2002.2.3

Lion S (2016) Moment equations in spatial evolutionary ecology. *Journal of Theoretical Biology*, **405**, 46–57. https://doi.org/10.1016/j.jtbi.2015.10.014

Louis P-Y, Nardi FR (Eds.) (2018) Probabilistic Cellular Automata: Theory, Applications and Future Perspectives. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-65558-1

Majumder S, Das A, Kushal A, Sankaran S, Guttal V (2021) Finite-size effects, demographic noise, and ecosystem dynamics. *The European Physical Journal Special Topics*, **230**, 3389–3401. https://doi.org/10.1140/epjs/s11734-021-00184-z

Manukyan L, Montandon SA, Fofonjka A, Smirnov S, Milinkovitch MC (2017) A living mesoscopic cellular automaton made of skin scales. *Nature*, **544**, 173–179. https://doi.org/10.1038/nature22031

Matsuda H, Ogita N, Sasaki A, Sato K (1992) Statistical Mechanics of Population: The Lattice Lotka-Volterra Model. *Progress of Theoretical Physics*, **88**, 1035–1049. https://doi.org/10.1143/ptp/88.6.1035

Mayor ÁG, Kéfi S, Bautista S, Rodríguez F, Cartení F, Rietkerk M (2013) Feedbacks between vegetation pattern and resource loss dramatically decrease ecosystem resilience and restoration potential in a simple dryland model. *Landscape Ecology*, **28**, 931–942. https://doi.org/10.1007/s10980-013-9870-4

Moreira J, Deutsch A (2002) Cellular Automaton Models of Tumor Development: A Critical Review. *Advances in Complex Systems*, **5**, 247–267. https://doi.org/10.1142/S0219525902000572

Muthukrishnan R, Lloyd-Smith JO, Fong P (2016) Mechanisms of resilience: empirically quantified positive feedbacks produce alternate stable states dynamics in a model of a tropical reef. *Journal of Ecology*, **104**, 1662–1672. https://doi.org/10.1111/1365-2745.12631

Pascual M, Roy M, Guichard F, Flierl G (2002) Cluster size distributions: signatures of self-organization in spatial ecologies. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **357**, 657–666. https://doi.org/10.1098/rstb.2001.0983

Petzoldt T, Rinke K (2007) simecol : An Object-Oriented Framework for Ecological Modeling in R. *Journal of Statistical Software*, **22**. https://doi.org/10.18637/jss.v022.i09

Pichon B, Donnet S, Gounand I, Kéfi S (2024) Estimating distances to tipping points from dryland ecosystem images. *bioRxiv*, 2024.02.20.581244. https://doi.org/10.1101/2024.02.20.581244

Sankaran S, Majumder S, Viswanathan A, Guttal V (2017) Patchiness and scale-free correlations: characterising criticality in ecosystems. *bioRxiv*, 233429. https://doi.org/10.1101/233429

Scanlon TM, Caylor KK, Levin SA, Rodriguez-Iturbe I (2007) Positive feedbacks promote power-law clustering of Kalahari vegetation. *Nature*, **449**, 209–212. https://doi.org/10.1038/nature06060

Schneider FD, Danet A, Génin A, Guttal V, Kéfi S, Majumder S, Sankaran S (2016) R-package caspr.

Shugart HH Jr, West DC (1981) Long-Term Dynamics of Forest Ecosystems: Computer simulation models, which allow for numerous seedlings and the long lives of large trees, predict how forests will respond to different management techniques. *American Scientist*, **69**, 647–652. https://www.jstor.org/stable/27850716

Siteur K, Liu Q-X, Rottschäfer V, van der Heide T, Rietkerk M, Doelman A, Boström C, van de Koppel J (2023) Phase-separation physics underlies new theory for the resilience of patchy ecosystems. *Proceedings of the National Academy of Sciences*, **120**, e2202683120. https://doi.org/10.1073/pnas.2202683120

Tucker GE, Hobley DEJ, Hutton E, Gasparini NM, Istanbulluoglu E, Adams JM, Nudurupati SS (2016) CellLab-CTS 2015: continuous-time stochastic cellular automaton modeling using Landlab. *Geoscientific Model Development*, **9**, 823–839. https://doi.org/10.5194/gmd-9-823-2016

Valiente-Banuet A, Ezcurra E (1991) Shade as a cause of the association between the cactus *Neobuxbaumia tetetzo* and the nurse plant *Mimosa luisana* in the Tehuacan Valley, Mexico. *The Journal of Ecology*, **79**, 961. https://doi.org/10.2307/2261091

Wilensky U (1999) NetLogo.

Wolfram S (1984) Cellular automata as models of complexity. *Nature*, **311**, 419–424. https://doi.org/10.1038/311419a0

Wootton JT (2001) Local interactions predict large-scale pattern in empirically derived cellular automata. *Nature*, **413**, 841–844. https://doi.org/10.1038/35101595

Xu C, Holmgren M, Van Nes EH, Maestre FT, Soliveres S, Berdugo M, Kéfi S, Marquet PA, Abades S, Scheffer M (2015) Can we infer plant facilitation from remote sensing? a test across global drylands. *Ecological Applications*, **25**, 1456–1462. https://doi.org/10.1890/14-2358.1

Yakowitz S, Gani J, Hayes R (1990) Cellular automaton modeling of epidemics. *Applied Mathematics and Computation*, **40**, 41–54. https://doi.org/10.1016/0096-3003(90)90097-M

Yassemi S, Dragicevic S, Schmidt M (2008) Design and implementation of an integrated GIS-based cellular automata model to characterize forest fire behaviour. *Ecological Modelling*, **210**, 71–84. https://doi.org/10.1016/j.ecolmodel.2007.07.020