# Easy, fast and reproducible Stochastic Cellular Automata with *chouca*

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

1 Copernicus Institute of Sustainable Development, Utrecht University, PO Box 80115, 3508 TC, Utrecht, The Netherlands.

2 Estación Costera de Investigaciones Marinas and Millenium Nucleus for Ecology and Conservation of Temperate Mesophotic, Reefs Ecosystems (NUTME) Pontificia Universidad Católica de Chile, Las Cruces, Chile.

3 International Master of Science in Marine Biological Resources (IMBRSea), Ghent University, Ghent, Belgium.

4 Centro Basal COPAS COASTAL, Universidad de Concepción

5 Instituto Milenio en Socio-Ecología Costera (SECOS), Santiago 8331150, Chile

6 Departamento de Ecología, Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile, Santiago 8331150, Chile

7 Advanced Conservation Strategies, Midway, UT 84049, USA

8 Center for Applied Ecology and Sustainability (CAPES) and Coastal Socio-Ecological Millenium Institute (SECOS), Pontificia Universidad Católica de Chile

# Abstract

* Stochastic cellular automata (SCA) are models that describe spatial ecological dynamics using a grid of cells that switch between discrete states over time, depending only on current states (Markov chain processes). 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 found in the literature are often ad-hoc, lacking performance and guarantees of correctness. More importantly, *de novo* implementation of SCA for each specific system and application 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*. which is an R package that translates intuitive SCA model declarations and expert-based assumptions about the state space system into compiled code and run simulations in a reproducible and efficient way.

* *chouca* supports a wide set of SCA along with deterministic cellular automata, with performance typically two to three orders of magnitude that of *ad hoc* implementations found in the literature, 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.

# Main text

## Introduction

The analysis of spatial patterns has proven essential to understand ecological system dynamics, and various modelling approaches help 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 (Louis & Nardi 2018; Wolfram 1984). Cellular automata are based on a grid of cells that switch over time between a finite number of states, which capture the entire state space of the system (all possible states assumed as relevant). Most often, SCA are considered to be distributed on 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. 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 typically depends 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 (Génin *et al.* in press; Muthukrishnan *et al.* 2016) 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 cases is a classical use-case of SCA for 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).

The popularity of SCA is probably related to 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 compute-intensive. Current approaches to do so efficiently rely on approximations, either assuming no spatial structure (mean field approximation) or simplifying the dynamics of the landscape to pairs of neighboring cells (“pair approximation”; (Iwasa 2000; Matsuda *et al.* 1992). However, these approximations are inaccurate when long-range correlations occur within a landscape (Iwasa 2000), or when the full simulated landscape is desired as a model output (*e.g.* to compute spatial metrics; Génin *et al.* 2018). In many cases, the explicit numerical simulation must be run, which is often done on small grids to reduce computation time. This is an issue as some theoretical results involving spatial structure or system stability have been shown to depend on this grid size (van de Koppel *et al.* 2011; Majumder *et al.* 2021). On top of performance issues, most ecological modeling work based on SCA comes with its own, ad-hoc 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.

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 $S_1 \ldots S_n$. Probabilities of transition are assumed to depend only on (i) the proportion of neighbors in each state, captured by the vector $\boldsymbol{q}=(q_1,\ldots,q_n)$, (ii) the proportion of cells in a given state in the whole landscape, $\boldsymbol{p}=(p_1,\ldots,p_n)$, and (iii) a set of constant model parameters $\boldsymbol{\theta}$. It is important to note that this excludes 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). Other types of SCA not fitting these constraints are those in which two cells swap their respective state, *e.g.* when modelling the movements of a predator in a landscape (Pascual *et al.* 2002). The probability of transition from state $i$ to $j$, $P(S_i \rightarrow S_j)$, 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:

$$P(S_i \rightarrow S_j)=a_0+g_1(q_1)+\ldots+g_n(q_n)+\xi(\boldsymbol{q},\boldsymbol{q})+\xi(\boldsymbol{q},\boldsymbol{p})+\xi(\boldsymbol{p},\boldsymbol{p}) \quad \text{(equation 1)}$$

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

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

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. 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")`.

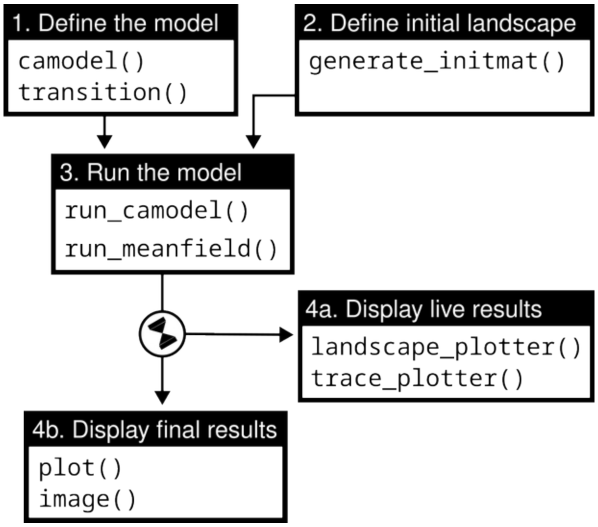

*Figure 1 Main tasks (boxes) of the chouca package, and their associated sets of functions*

## Using *chouca*: a simple model

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 in *chouca* 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 $\alpha\, q_{mussel}$, where $\alpha$ is a productivity rate, and $q_{mussel}$ is the proportion of cell neighbors in the mussel state. Such transition is defined by the following function 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"]` could have been 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 $\delta$, 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 (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),
  wrap = TRUE,
  neighbors = 4
)
```

The call to the `camodel()` function above estimates the constants in the functional form described in equation 1 to fit 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)
```

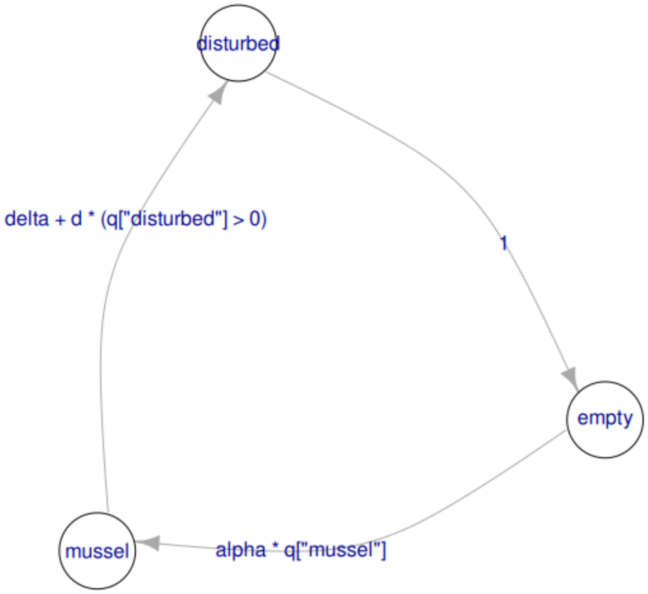

*Figure 2 Structure of the mussel bed model displayed as a graph, in which nodes are states, and arrows represent the possible transitions with the expression used to compute their probabilities*

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 = 0.5, mussel = 0.8),
                                   nr = 64, nc = 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 (respectively):

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

By default, *chouca* uses a C++ backend, which is reasonably fast. Performance can be increased 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 allows reaching typical simulation speeds above 5000 iterations.s$^{-1}$ on a typical 128 x 128 grid (Figure 3), 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 list controls 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 is in the help page `?run_camodel`). 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 spatial statistics (*e.g.* autocorrelation) on the 2D landscape as the simulation is running, which we illustrate below.

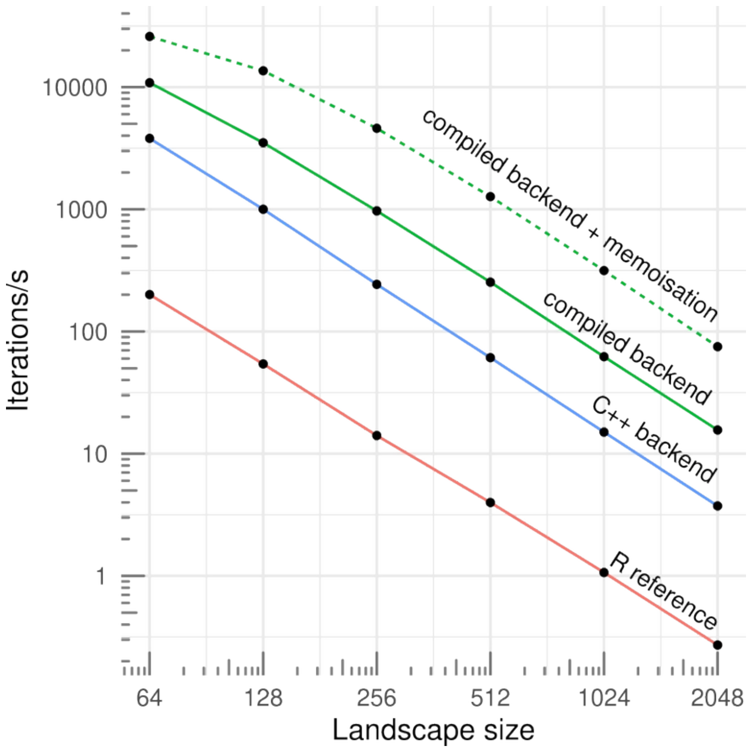

*Figure 3 Simulation speeds for a set of three simple ecological models (2-3 states and 3-4 transitions) according the the grid size, for a pure-R implementation (R reference; Schneider et al. 2016), and three backends provided by* chouca *(blue and green lines).*

## 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 (e.g. Hartig et al. 2011 and section below), but the qualitative comparison and visual exploration of model dynamics remains an essential phase for 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 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 may be difficult (for ecologists) 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 model behavior along a range of parameter values.

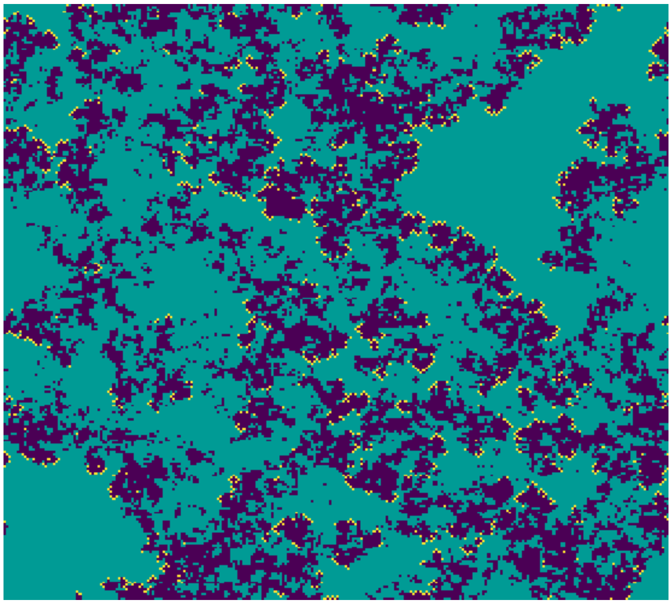

*Figure 4 Landscape patterns of the host-parasitized-empty model (respectively green, yellow and blue) as displayed on screen*

## 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, plants often facilitate each other, 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 (Chen *et al.* 2022; Xu *et al.* 2015). 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 (Kéfi *et al.* 2011; Siteur *et al.* 2023), and linking the observed changes in those metrics to theoretical results (Kéfi *et al.* 2011; Scanlon *et al.* 2007). 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 $\beta$. $\beta$ 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 can be 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 256×256 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 the final landscape, we computed the distribution of pairs, which summarizes all the possible pairs of neighboring cells present in the landscape $\boldsymbol{n_{obs}}=(n_{p,0},n_{p,p},n_{0,0})$. To estimate the likelihood $P(\boldsymbol{n_{obs}}|d,\beta)$, we assume 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 $\beta$, and can be estimated by simulating the landscape with these parameter values. This likelihood estimate formally quantifies the agreement between model and data, and allows exploring the parameter space in terms of $d$ and $\beta$ to find the best parameter combinations given the observed patterns.

We find that this approach can recover the parameter values used for $d$ and $\beta$ , with only one global maximum in the likelihood function (Figure 5). 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 uncertainty 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 exercise requires running around 5 000 simulations. This would be unrealistic with usual SCA implementations, but takes roughly three minutes with *chouca* on a 2020 desktop computer (12 cores). This way of calibrating models has seldom been used in ecology – a fast SCA implementation is essential to make it more accessible.

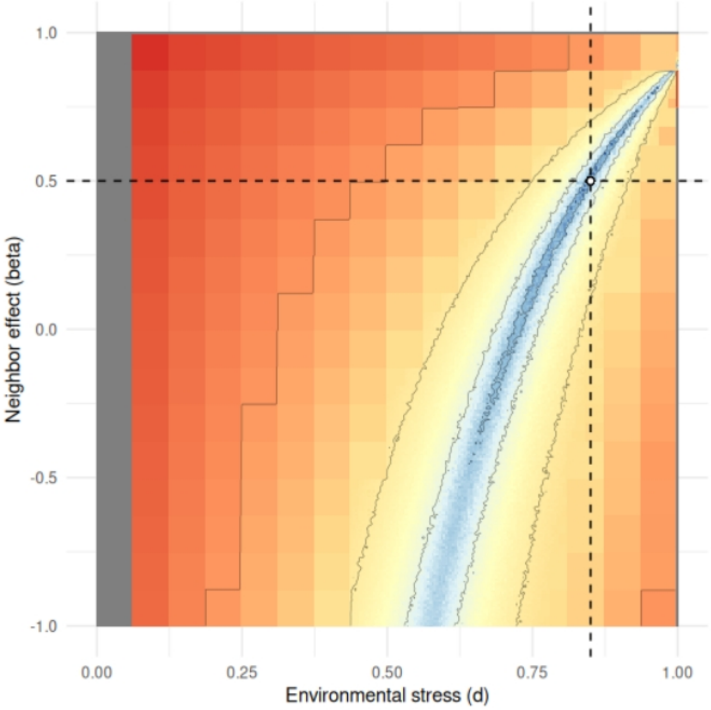

*Figure 5 Likelihood surface as a function of the two model parameters d and β. Blue values indicate parameter combinations with high likelihood. The white dot and lines indicate the real parameter values.*

## Concluding remarks

*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 efficient simulations for the type of SCA it supports. 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 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* will be available on CRAN, and welcomes comments, feedback and bug reports on its home page at https://github.com/alexgenin/chouca.

## References

Chen, B.J.W., Teng, S.N., Zheng, G., Cui, L., Li, S., Staal, A., *et al.* (2022). Inferring plant–plant interactions using remote sensing. *Journal of Ecology*, 110, 2268–2287.

Génin, A., Majumder, S., Sankaran, S., Danet, A., Guttal, V., Schneider, F.D., *et al.* (2018). Monitoring ecosystem degradation using spatial data and the R package spatialwarnings. *Methods Ecol Evol*, 9, 2067–2075.

Génin, A., Navarrete, S.A., Garcia-Mayor, A. & Wieters, E.A. (in press). Emergent spatial patterns can indicate upcoming regime shifts in a realistic model of coral community. *The American Naturalist*.

Guichard, F., Halpin, P.M., Allison, G.W., Lubchenco, J. & Menge, B.A. (2003). Mussel disturbance dynamics: signatures of oceanographic forcing from local interactions. *The American Naturalist*, 161, 889–904.

Hartig, F., Calabrese, J.M., 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.

Heinonen, T. & Pukkala, T. (2007). The use of cellular automaton approach in forest planning, 37.

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

Kéfi, S., Rietkerk, M., Alados, C.L., Pueyo, Y., Papanastasis, V.P., ElAich, A., *et al.* (2007). Spatial vegetation patterns and imminent desertification in Mediterranean arid ecosystems. *Nature*, 449, 213–217.

Kéfi, S., Rietkerk, M., Roy, M., Franc, A., de Ruiter, P.C. & Pascual, M. (2011). Robust scaling in ecosystems and the meltdown of patch size distributions before extinction: Patch size distributions towards extinction. *Ecology Letters*, 14, 29–35.

van de Koppel, J., Gupta, R. & Vuik, C. (2011). Scaling-up spatially-explicit ecological models using graphics processors. *Ecological Modelling*, 222, 3011–3019.

Lanzer, A.T.S. & Pillar, V.D. (2002). Probabilistic cellular automaton: model and application to vegetation dynamics. *Community Ecology*, 3, 159–167.

Louis, P.-Y. & Nardi, F.R. (Eds.). (2018). *Probabilistic Cellular Automata: Theory, Applications and Future Perspectives*. Emergence, Complexity and Computation. Springer International Publishing, Cham.

Majumder, S., Das, A., Kushal, A., Sankaran, S. & Guttal, V. (2021). Finite-size effects, demographic noise, and ecosystem dynamics. *Eur. Phys. J. Spec. Top.*, 230, 3389–3401.

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.

Mayor, Á.G., 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 Ecol*, 28, 931–942.

Muthukrishnan, R., Lloyd-Smith, J.O. & 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.

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.

Petzoldt, T. & Rinke, K. (2007). **simecol**: An Object-Oriented Framework for Ecological Modeling in *R. J. Stat. Soft.*, 22.

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

Scanlon, T.M., Caylor, K.K., Levin, S.A. & Rodriguez-Iturbe, I. (2007). Positive feedbacks promote power-law clustering of Kalahari vegetation. *Nature*, 449, 209–212.

Siteur, K., Liu, Q.-X., Rottschäfer, V., van der Heide, T., Rietkerk, M., Doelman, A., *et al.* (2023). Phase-separation physics underlies new theory for the resilience of patchy ecosystems. *Proc. Natl. Acad. Sci. U.S.A.*, 120, e2202683120.

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.

Wilensky, U. (1999). *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, 311, 419–424.

Wootton, J.T. (2001). Local interactions predict large-scale pattern in empirically derived cellular automata. *Nature*, 413, 841–844.

Xu, C., Holmgren, M., Van Nes, E.H., Maestre, F.T., Soliveres, S., Berdugo, M., *et al.* (2015). Can we infer plant facilitation from remote sensing? a test across global drylands. *Ecological Applications*, 25, 1456–1462.

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*, 71–84.