/Revision round *#1*/

Decision for round #1 : */Revision needed/*

a useful software package to better position in the current literature

----

As you will see in the attached reviews, the two reviewers and myself all agree about the usefulness of your software package for the community, but we noticed a few issues that would need to be addressed in a revision before envisaging a formal recommendation. First, as pointed out by the first reviewer, the width of the cellular automata literature could be better acknowledged and, therefore, so could be the limitations of the package. Second, as pointed out by the second reviewer, further discussing existing packages on other platforms, such as Python, would help better position your package. Finally, to follow-up on a comment from reviewer #1 about biological applications, I would like to add that the example of lizard scales is also one of the most inspiring ones I have seen of late (Manukyan et al 2017 Nature, DOI: 10.1038/nature22031).

Thank you for your constructive feedback on our manuscript, and coordinating its review. We paid specific attention to the points raised, with a specific effort towards better acknowledging the breadth of the CA literature (to the best of our ability given its size), along with the limits of the package.

Regarding these limits, please note that it is the nature of all canned software to have limitations – it is, in fact, such restrictions that allow us to provide optimized code for a given model. Many CA engines are essentially wrappers to an “update function”, which is a language function that provides the new state of the grid or a cell, depending on the current state – this is extremely flexible, but all performance concerns are the responsibility of the user (*e.g.* caspr, cellpylib, links below). Here, we turn this around and restrict the set of models that can be implemented, but the package provides optimized code for a specific model definition.

We paid specific attention to better stating the limits of the package in our revision, and what may/may not be changed in future versions. We take here stochastic CA in the restricted sense of “locally-interacting markov chains”, and some improvements suggested by reviewer #2 fall within that definition and will or are currently being implemented (e.g. different types of grids or neighborhoods). Other ideas probably fall out of that scope or appear difficult to implement without large changes (3D grids, continuous cell values, cells swapping states), though we acknowledge the broader set of processes these improvements could capture.

Regarding other software packages (Comments by Reviewer #1), it is important to note that while many exist to simulate cellular automata (such as CellPyLib), very few of those packages apply to *stochastic* cellular automata. This is an essential distinction because deterministic CA are fundamentally different models. This being said, a remarkable exception that implements asynchronous stochastic cellular automata may be CellLab-CTS (python), which elegantly allows defining probabilistic transitions between pairs of cells, and that we mention in our revision.

You will find below a point-by-point response to the comments received on the manuscript, along with the specific lines that have been changed. We hope this addresses the concerns you and the reviewers raised, and look forward to hearing back from you.

Many thanks,

Alexandre Génin, on behalf of all the co-authors

Links:
CellLab-CTS (python) : https://landlab.readthedocs.io/en/latest/user_guide/cell_lab_user_guide.html
CellPyLib (python): https://cellpylib.org/
Caspr (R): https://github.com/fdschneider/caspr

/Minor suggestions/

l.28, 29, 30: repetition of "scales"
l.29: "Markovian" might be a bit technical word for the first sentence of a ecology abstract.
l.35: mention reproducibility?
l.35: delete "more importantly"?
These changes have been made in the revision.

l.84: "computationally" instead of "compute"?
l.86: mention moment equations rather than just "pair approximation"? (see also the review by Lion 2016, JTB 10.1016/j.jtbi.2015.10.014)
These changes have been implemented in the revision. Thank you for mentioning this review, which was very interesting and very much on point.

l.98: I would start a new section here because this seems to be more about "The Model" rather than the "Introduction".
We do so in the revision (new section “Supported models”).

l.123: Perhaps mention Rcpp at this stage?
This figure presents the main tasks associated to the package, so we are unsure whether mentioning Rcpp there is valuable – it is mentioned as a key element to performance though further down (l. 204)

l.124: This could be rephrased a bit into a classical "Results" section or "Illustration"
We settled on “Example applications” in the revision.

l.195: Write "This can be improved by" instead of "improved on by"?
This has been fixed in the revision.

l.213: Schneider et al (2016) does not seem to be in the bibliography.
This reference (to software) has been added now.

l.217: The previous section already seemed to be an illustration example. Perhaps merge the two could help?
This has been done in the revision, with all examples put in the same section (“Example applications”, with three subsections).

/by ***Samuel Alizon***, 13 May 2024 16:31/

Review by anonymous reviewer 1, 11 May 2024 11:16

- Comments & Suggestions

In general, this is a well-written manuscript that provides a comprehensive description of the SCA application to landscape pattern simulations, based on the Chouca package. The examples used in the manuscript effectively demonstrate the utility of the package in ecological modelling.

My primary suggestion for improvement would be to include a more detailed comparison with other existing tools (e.g., CellPyLib for Python) in the discussion. This would more distinctly highlight Chouca's unique features and its advantages and disadvantages.

Thank you for your feedback. CellPyLib, and many other CA engines, do not allow for probabilistic transitions, making the comparison difficult. We do emphasize CellLab-CTS though in the revision, which is concerned with stochastic cellular automata.

Review by Broder Breckling, 13 May 2024 14:24

Génin et al present a framework to implement a kind of simplistic cellular automata with probabilistic state transitions. They use illustrative ecological examples to show the functionality. Because they apply R as a computational framework, it can be expected that it makes the access for users easier compared to genuine programming. However, the approach is syntactically rather limited and excludes important ecological interaction types, e.g. it operates on 2d rectangular grids only and excludes direct cell-to-cell relation that would allow 1:1 mutual state change of cells. Besides specific application cases, its benefit could be in the introduction to spatial modelling in a didactical context. For research applications, in most practical cases a higher flexibility and complexity in rule-specification might be required.

We thank the reviewer for his review and feedback on the package. We use, indeed, a somewhat restricted definition for stochastic cellular automata and view it more from the point of view of locally-interacting markov chains. We hope to relax some of the constraints the package currently has (grid type, neighborhood size, isotropy), but other suggestions fall out of scope of this definition (e.g. cells swapping state). We point to other solutions for such cases (e.g. CellLab-CTS).

In our revision, we paid specific attention to making clearer what types of models are/are not supported, and what limitations may be lifted in future versions of the package.

In the following, comments to particular statements in the text are added. Because the paper does not provide line numbers, the commented sentences are copied below.

Specific comments

The paper states "Stochastic cellular automata (SCA) are models that describe spatial dynamics using a grid of cells that switch between discrete states over time, depending only on the current state

(Markovian processes).”

Comment

This is a rather narrow specification which excludes an application for many practical requirements. If for the characterisation of a cell’s state continuous variables are excluded, many relevant processes cannot be modelled e.g. the concentration of a substance, continuous variation in pressure, temperature, etc. at a particular grid position. This is in particular relevant for ecological applications.

It is true that our work does not allow for representing continuous cell states – we think assuming discrete cell states seems to be a reasonable assumption given the literature on SCA, but this is definitely a limitation indeed.

The paper states “de novo implementation of SCA for each specific system and application represents a major barrier for many practitioners.”

Comment
It is questionable if this is actually a barrier. Even without object-oriented programming, it is not such a challenge to develop a CA. For a rectangular CA with a von Neumann neighbourhood, two nested FOR-loops together with conventional calculation would do as a main component to set the layout for an iteration cycle.

It is true that the implementation of an SCA is not, in the absolute sense, a hard task in terms of pure programming skills – though there are some pitfalls that need to be taken care of (e.g. probabilities above 1, handling transitions to multiple states with equal probability). This being said, reducing this work to essentially zero with a declarative interface frees time for other tasks. An important one we envision is to compare models. It is common, for example, to compare statistical models in terms of AIC/BIC or other metrics measuring the agreement between model and data, in order to test hypotheses on the existence of ecological processes. This, however, is rarely done (to our knowledge) with semi-mechanistic models like cellular automata, and we can conjecture this is because it is non-trivial to write such models.

The paper states
“we built chouca, an R package that translates intuitive SCA model definitions into compiled code,”
Comment
this may be a major advantage compared to interpreted languages. The authors present a figure (Fig. 3) which compares runtimes of different grid size and indicate a considerable efficiency.

The paper states
“ad hoc implementations found in the literature,”
Comment
here, a reference would be useful to clarify which implementations the authors have in mind

We rephrased this sentence to remove the mentions to existing literature (l. 91). Many, if not most SCA implementations are done from scratch for the purpose of one or a few specific paper(s), so it is hard to identify a single reference here.

The paper states
“Conway’s game of life,”

Comment
a reference would be nice

We know refer to the original Scientific American article, and an introductory chapter on the topic that we found useful (l. 64-65).

The paper states
“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.”

Comment
Frequently, the change of the cell’s states in CAs are considered to depend on the state of the cell itself and its neighbours. The specification here, expands that and offers to operate on two levels of neighbourhood – a local and a global one. This is possible, however, the evaluation of the entire grid state to specify a new cells state is usually not required as a CA standard. Nevertheless, under particular condition, this could be an interesting option.

From our knowledge of the literature, these two scales are typical when modelling the dynamics of sessile organisms (plants, forest, etc.), as a split is often done between processes that occur locally (fires, clone growth, etc.) or globally (i.e. homogeneously over the grid, e.g. dispersal of propagules).

The paper states
“In many cases, the explicit numerical simulation must be run, which is often done on small grids to reduce computation time.”
Comment
What do the authors understand as a “small grid”? Since computational potentials expand with technical developments, this statement is relative and tends to lose relevance. Is the grid size stated below (p.7: “a typical 128 x 128 grid (Figure 3)” a small one?

A grid of 100x100 is typical in our experience. We rephrased both sentences to remove all references to sizes or small/large grids (see l. 91).

The paper states
“This opens the possibility of errors in code and often makes it difficult to reproduce model simulations.”
Comment
this is a widespread issue throughout ecological modelling. It may be a useful contribution of this approach, if for this specific segment in modelling, result reproduction may become easier.

Thank you for this comment, it is, indeed, one of the goal of this work.

The paper states
The R package chouca works with 2-dimensional rectangular grids of cells (a “landscape”).
Comment
In an upcoming version, the authors should expand it to 3d grids, then allowing to represent also waterbodies or processes referring to atmospheric layers. A further step then could be to facilitate the use of triangular or hexagonal grids. To have an alternative to rectangular grids would allow for tests of the extent of discretisation artefacts – which are quite relevant for rectangular grids with regard to

ecological processes.

The idea to extend models to other types of grids is interesting and may be included in a future version of the package – this is a natural extension of the existing codebase. However, extending it to 3D grids would probably require large amounts of work so this may be difficult in the near future.

The paper states
“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)” … “In practice, this functional form is flexible enough to approximate the probabilities of transition of many ecological models”
Comment
this excludes a very large extend (if not the majority) of potential ecological CA applications. If not even the change of a state’s cell with its neighbour could be represented, then the whole class of pattern generating processes like Diffusion Limited Aggregation as well as a majority of trophic interaction modelling would be excluded.

This is correct – in the revision, we tried to provide a clearer picture of what models can or cannot be used with *chouca* (see ll. 105-116) and our comments at the beginning of this review.

The paper states
“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”
Comment
even more often, in arid systems plants limit each other through competition for water access and give rise to gaps in vegetation cover. Facilitation is a process that helps in the survival of seedlings and is usually only a part of the relevant interactions. To describe plant dynamics under arid conditions might be more convincing by using “inhibition” rather than “facilitation”…

Thanks for pointing this out – we rephrased this sentence to highlight the fact that plant-plant interactions in those systems are often a balance of negative and positive effects (ll. 271-274).