Fragmented Landscape Generator (`flsgen`): a neutral landscape generator with control of landscape structure and fragmentation indices

Dimitri Justeau-Allaire[1,2*], Grégoire Blanchard[2], Thomas Ibanez[2], Xavier Lorca[4], Ghislain Vieilledent[2,3], & Philippe Birnbaum[1,2,3].

1. *Institut Agronomique néo-Calédonien (IAC), Nouméa, New Caledonia*

2. *AMAP, Univ Montpellier, CIRAD, CNRS, INRAE, IRD, Montpellier, France*

3. *CIRAD, UMR AMAP, Montpellier, France*

4. *Centre de Génie Industriel, IMT Mines Albi, Albi, France*

*Corresponding author: dimitri.justeau@gmail.fr

Abstract

1. Neutral landscape models have many applications in ecology, such as supporting spatially-explicit simulations, developing, and evaluating landscape indices. However, current approaches provide few options to produce large landscapes with controlled composition and fragmentation indices.

2. We introduce `flsgen` (Fragmented Landscape Generator), a new neutral landscape generator that address this limitation by providing a high level of control over 14 landscape indices. The main novelty of `flsgen` is the decomposition of landscape generation into two steps: the solving of a constraint satisfaction problem and the generation of a landscape raster with a stochastic algorithm. The latter relies on a continuous environmental gradient that influences the landscape's spatial configuration.

3. `flsgen` can generate fine-grained artificial landscapes in small amounts of time, which makes it suited to produce large landscape series systematically. We demonstrate the features of `flsgen` through three illustrative use cases.

4. `flsgen` is a practical and efficient tool that expand the current possibilities of neutral landscape models and widen their potential applications. To facilitate its uptake, `flsgen` is available as free and open-source software through a Java API, a command-line interface, or an R package.

Keywords: Artificial landscape generation; Neutral landscape; Landscape ecology; Habitat fragmentation; Constraint programming; Landscape indices.

1 Introduction

Landscape spatial patterns are known to influence ecological processes (Turner, 1989). For instance, the size and distribution of habitat patches can influence species immigration and extinction which in turn affect diversity patterns. However, such relations between patterns and processes are still not well understood and likely to differ among species and ecosystems (Rutledge, 2003; Frazier and Kedron, 2017). To address this challenge, researchers often rely on landscape indices (Ibanez et al., 2017; Cuervo and Møller, 2020), computer simulations (Bowers et al., 1996; Wiegand et al., 2005; Rahimi et al., 2021), or experiments on controlled landscapes (Collins and Barrett, 1997; Seibold et al., 2017; With and Payne, 2021).

As landscape-level experiments are often not feasible, several artificial landscape models have been developed to support such studies. They can be separated into two categories: *process-based* models and *neutral models* (or *pattern-based*) (van Strien et al., 2016). In the first category, landscapes are generated according to spatial patterns that are associated with ecological or anthropogenic processes (e.g. Gaucherel et al., 2006; Pe'er et al., 2013; Dislich et al., 2018). In the second category, landscape generation relies on random spatial processes, including cellular-automata (e.g. Soares-Filho et al., 2002), fractal geometry (e.g. Gardner, 1999; Hargrove et al., 2002), and multi-objective optimization algorithms (e.g. van Strien et al., 2016). In such neutral models, landscape composition and fragmentation can be controlled through parameters that are specific to the random spatial algorithms, such as the H parameter (or roughness factor) which is used in the *diamond-square* (or *midpoint displasment*) algorithm to control the level of "fragmentedness" (Fournier et al., 1982; Neel et al., 2004; Cambui et al., 2015).

However, as pointed out by van Strien et al. (2016), such parameters do not reflect how real landscapes are evaluated in landscape ecology, where various metrics are available to describe the composition and configuration of a given landscape. This can be problematic to address research questions involving a systematic exploration of landscape indices. In their software *Landscape Generator* (`LG`), van Strien et al. (2016) addressed this limit of neutral landscape models, making it possible to generate artificial landscapes using the same parameters used to evaluate real landscapes. In `LG`, the user defines target values to control patch and class-level landscape indices such as the number of patches, the total habitat amount, and patch-level indices such as patch area, or patch maximum perimeter. In addition, van Strien et al. (2016) presented some potential improvements to increase the control over generated landscapes. Notably, they suggested integrating more landscape indices as user targets, such as the largest patch index. Moreover, they recognized that the computation time of `LG` needs to be improved. Indeed, `LG` relies on a multi-objective optimization algorithm which can take several hours to generate 50$\times$50 pixels landscapes and increases exponentially with increasing landscape size, making it unsuited to generate large landscapes and large series of landscapes. Furthermore, `LG` does not provide targets over advanced fragmentation indices, such as the *effective mesh size* (e.g Jaeger, 2000). This index, which is based on the probability that two random points are located in the same patch, is widely used in fragmentation studies (e.g. Schmiedel and Culmsee, 2016; Babí Almenar et al., 2019; Cuervo and Møller, 2020) and would be a great asset as a user-target in neutral landscape models.

In this article, we address some of LG's limitations with *Fragmented Landscape Generator* (`flsgen`), a new neutral landscape generator that offers a high level of control over landscape composition and fragmentation. Specifically, `flsgen` offers an expressive control over 14 landscape indices (see Table 1), including advanced fragmentation indices such as the effective mesh size. Although targets focus on composition and fragmentation, the spatial configuration of landscapes can be controlled with continuous environmental gradients. The main technical novelty of `flsgen` is the decomposition of landscape generation into two distinct processes: the identification of suitable landscape structures by solving a constraint satisfaction problem with a constraint programming (CP) solver, and the spatial landscape generation with a stochastic algorithm. This approach allows `flsgen` to generate landscapes with millions of cells, hundreds of patches, and several land-use classes within seconds, which makes it suited for large-scale experiments and analysis. `flsgen` is available as free and open-source software through a Java API, a command-line interface, and an R package.

Name	Abbreviation	Level	Unit
Patch area	AREA	class	cell surfaces
Mean patch area	AREA_MN	class	cell surfaces
Total class area	CA	class	cell surfaces
Proportion of landscape	PLAND	class	percentage
Number of patches	NP	class	unitless
Patch density	PD	class	patches per cell surface
Smallest patch index	SPI	class	cell surfaces
Largest patch index	LPI	class	cell surfaces
Effective mesh size	MESH	class	cell surfaces
Splitting index	SPLI	class	unitless
Net product	NPRO	class	(cell surfaces)^2
Splitting density	SDEN	class	(cell surfaces)$^{-1}$
Degree of coherence	COHE	class	probability (in [0,1])
Degree of landscape division	DIVI	class	probability (in [0,1])

Table 1: Currently available user targets. The first group contains simple indices (McGarigal et al., 2012), and the second group contains advanced fragmentation indices (Jaeger, 2000).

2 Overview of `flsgen`

`flsgen` consists of two main components: (i) a constrained landscape structure solver, `flsgen structure`, which produces non-spatially-explicit patch area distributions satisfying all user targets, and (ii) a spatially-explicit stochastic algorithm, `flsgen generate` which generates neutral landscapes satisfying predefined patch area distributions and relies on continuous environmental gradients to control spatial configuration. These components can be used independently, or the first one can serve as input for the second. Additionally, landscape structures can be extracted from real landscapes to recreate real composition patterns. Figure 1 summarizes `flsgen`'s workflow, and Table 1 depicts available user targets. The area unit for `flsgen` targets is the cell surface, and geographical attributes (spatial extent, coordinate reference system, resolution) of the produced rasters can specified by the user. The dimensions of generated landscapes are either specified by the user or defined through a mask raster. Also note that `flsgen` allows setting a target on the proportion of landscape unoccupied by the focal classes (NON_FOCAL_PLAND). This space corresponds to what we called the *non-focal* class, that is the matrix surrounding focal classes.

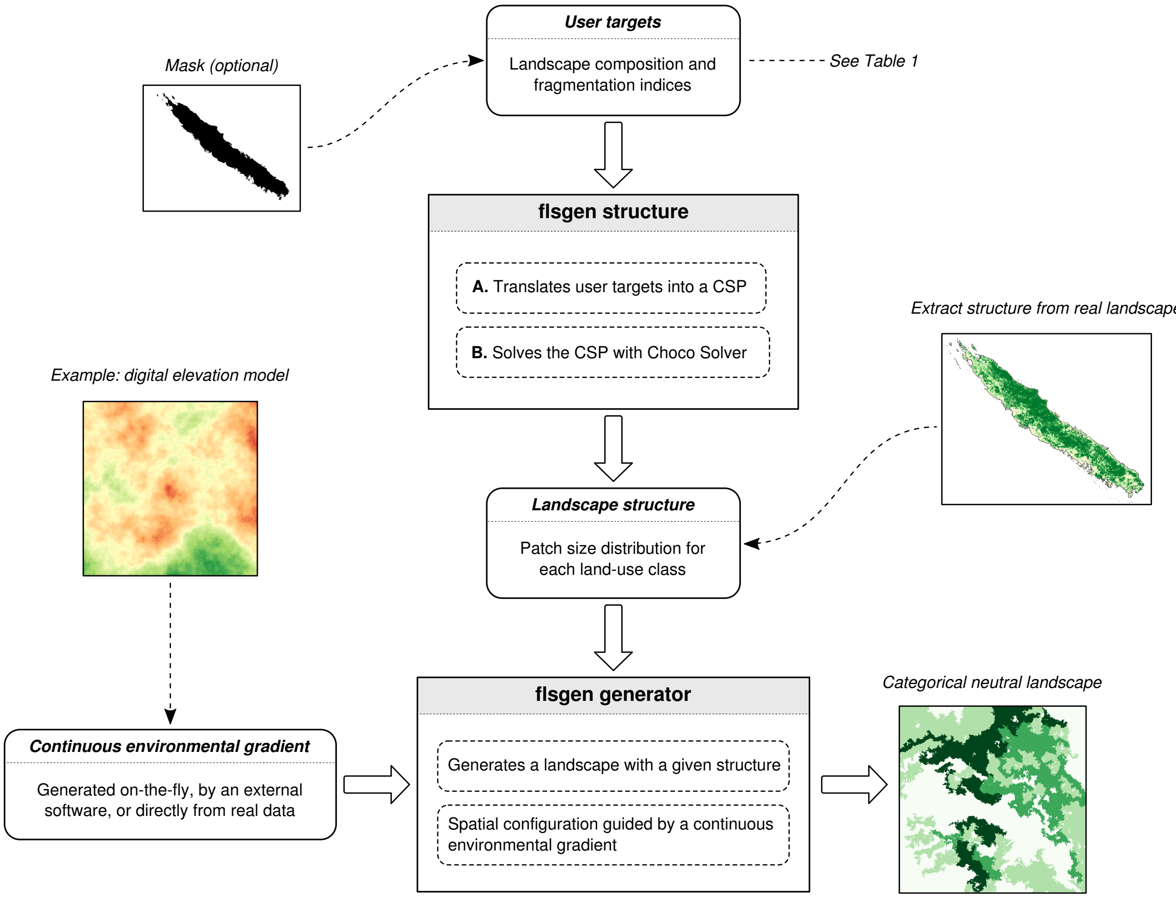

Figure 1: `flsgen` workflow: landscape structures (non-spatially-explicit) satisfying user targets are generated with `flsgen structure`, whose outputs are used by `flsgen generator` to generate spatially-explicit landscape rasters. The generation algorithm relies on a continuous environmental gradient, which can either be given as input or generated on-the fly as a fractal terrain. User targets can include a mask, and landscape structures can also be extracted from real landscapes.

2.1 Description of the landscape structure solver

The first main component of `flsgen` is also the most distinctive from classical neutral landscape generation approaches. It consists of a constrained landscape structure solver, `flsgen structure`. Given a set of focal land-use classes and user targets, it is able to identify a set of non-spatially explicit landscape structures (i.e. a patch size distribution for each focal land-use class) such that *all* user targets are satisfied. If the targets do not admit any feasible landscape structure (e.g. two distinct classes both occupying 60% of the landscape), `flsgen structure` is able to detect such cases and inform the user that targets cannot be satisfied. Depending on user-targets, there may be thousands of suitable landscape structures, consequently, it is up to the user to specify how many solutions are desired. Note that it is possible to diversify the solutions (see *Frequently asked questions* in Supplementary Information) The implementation is based on a constraint satisfaction problem (CSP). In a nutshell, a CSP is a mathematical problem where, given a set of variables $\mathcal{X} = \{X_1, ..., X_n\}$ taking their values in the domains represented by $\mathcal{D} = \{D_1, ..., D_n\}$, the aim is to find a set of values $\{v_1 \in D_1, ..., v_n \in D_n\}$ satisfying a set of constraints denoted by $\mathcal{C}$. The CSP solved in `flsgen structure` expresses as follows. Given:

- L_S the total landscape area;
- N the number of landscape classes;
- $\underline{\mathrm{NP}}_1, ..., \underline{\mathrm{NP}}_N$ the minimum number of patches for each class;
- $\overline{\mathrm{NP}}_1, ..., \overline{\mathrm{NP}}_N$ the maximum number of patches for each class;
- $\underline{\mathrm{AREA}}_1, ..., \underline{\mathrm{AREA}}_N$ the minimum patch area for each class;
- $\overline{\mathrm{AREA}}_1, ..., \overline{\mathrm{AREA}}_N$ the maximum patch area for each class;
- $\underline{\mathrm{CA}}_1, ..., \underline{\mathrm{CA}}_N$ the minimum total area for each class;
- $\overline{\mathrm{CA}}_1, ..., \overline{\mathrm{CA}}_N$ the maximum total area for each class;
- $\underline{\mathrm{NPRO}}_1, ..., \underline{\mathrm{NPRO}}_N$ the minimum net product[1] for each class;
- $\overline{\mathrm{NPRO}}_1, ..., \overline{\mathrm{NPRO}}_N$ the maximum net product for each class;

Find a patch area distribution $P_i = \{\mathrm{AREA}^i_1, ..., \mathrm{AREA}^i_{\mathrm{NP}_i}\}$ (with NP_i the variable representing the number of patches of class i and AREA^i_j the variable representing the area of patch j from class i) for each landscape class i such that:

[1] i.e. the sum of squared patch areas (Jaeger, 2000)

$$\underline{\mathrm{NP}}_i \leq \mathrm{NP}_i \leq \overline{\mathrm{NP}}_i \quad \text{for all } i \in [1, N]; \tag{1}$$

$$\underline{\mathrm{AREA}}^i_j \leq \mathrm{AREA}^i_j \leq \overline{\mathrm{AREA}}^i_j \quad \text{for all } i \in [1, N] \text{ and for all } j \in [1, \mathrm{NP}_i]; \tag{2}$$

$$\underline{\mathrm{CA}}_i \leq \sum_{j \in [1, \mathrm{NP}_i]} \mathrm{AREA}^i_j \leq \overline{\mathrm{CA}}_i \quad \text{for all } i \in [1, N]; \tag{3}$$

$$\underline{\mathrm{NPRO}}_i \leq \sum_{j \in [1, \mathrm{NP}_i]} (\mathrm{AREA}^i_j)^2 \leq \overline{\mathrm{NPRO}}_i \quad \text{for all } i \in [1, N]; \tag{4}$$

$$\sum_{i \in [1, N]} \mathrm{CA}_i \leq L_S. \tag{5}$$

Constraints (1), (2), (3), and (4) respectively ensure that the number of patches (NP), patch areas (AREA), total class area (CA), and the net product (NPRO) take their values within specified bounds. Constraint (5) ensures that the landscape configuration does not exceed the total landscape area. In this CSP, constraining NP, AREA, CA, and NPRO is sufficient to allow any other index from Table 1 to be set as a target, as all of these indices are proportional to either NP, AREA, CA, or NPRO. For example, if we want to enforce $\mathrm{PLAND}_i \geq \underline{\mathrm{PLAND}}_i$, we just need to set $\underline{\mathrm{CA}}_i = \frac{\mathrm{PLAND}_i L_s}{100}$. Similarly, a minimum effective mesh size $\underline{\mathrm{MESH}}_i$ for a class i can be set as target by setting $\underline{\mathrm{NPRO}}_i = \underline{\mathrm{MESH}}_i \times L_s$ (see Jaeger, 2000). All of these operations are hidden to users, who only need to set their targets for any of the indices in Table 1. To solve this CSP, `flsgen structure` relies on *Choco solver* (Prud'homme et al., 2017), an open-source Java Constraint Programming (CP) solver, which provides an exact solving engine based on artificial intelligence techniques such as automated reasoning, constraint propagation and search heuristics (Rossi et al., 2006).

2.2 Description of the neutral landscape generator

To generate spatially-explicit landscape satisfying landscape structures generated by `flsgen structure`, we implemented `flsgen generate`, a stochastic neutral landscape generator. Using a stochastic algorithm cannot guarantee that a feasible landscape will be found, neither that a spatial embedding of the input structure exists. However, generating a 2D raster landscape with a predefined structure is equivalent to solving a polyomino packing problem, which is known to be NP-Complete even for small shapes (Brand, 2017). Consequently, using an exact approach for this step would likely slow down the generation and limit the output spatial resolution. In practice, our approach is efficient for most cases, and is more likely to fail when focal classes occupy more than 90% of the total landscape area.

The main input of our algorithm is a landscape structure with N landscape classes and a set of patch area distributions $P = \{P_1, ..., P_N\}$ such that for any landscape class i, $P_i = \{\text{AREA}_1^i, ..., \text{AREA}_{\text{NP}_i}^i\}$ with NP_i the number of patches in class i and AREA_j^i the area of patch j in class i. To generate a landscape, the algorithm iteratively tries to fill an empty landscape with each class (see Algorithm 1 in Supplementary Information). Given a class, it iteratively constructs each patch specified in the structure by first randomly selecting an available cell in the landscape, and then by randomly adding available cells that are in the neighbourhood of already selected cells (see Algorithm 2 in Supplementary Information). A cell is considered available if it is not already assigned to a landscape class and if it is not in the buffer of another patch of the same class. The width of patch buffers represents the minimum distance between two patches of the same class and is specified by the user with the d_b parameter. The selection of a cell is affected by the input continuous environmental gradient, also named the *terrain*, according to the *terrain dependency* parameter t_d. It corresponds to one minus the proportion of neighbouring cells with the lowest value in the terrain that can be selected (see $filter$ function of the Algorithm 2 in Supplementary Information). Setting $t_d = 1$ forces the algorithm to always select the available cell with the lowest value, whereas setting $t_d = 0$ makes the algorithm insensitive to the environmental gradient.

2.3 Distribution

The software `flsgen` is distributed as an open-source software under the GNU GPL3 licence. Source code and downloads are available in GitHub. The software can be used as a Java API, an R package, or through a command-line interface (CLI).

Java API (`https://github.com/dimitri-justeau/flsgen`): The three components of `fslgen` were developed in Java. The Java API of `flsgen` is then its native API and offers a great flexibility. Notably, using `flsgen` from Java offers a full access to the Choco solver library, which makes it appropriate for advanced uses.

R package (`https://github.com/dimitri-justeau/rflsgen`): To facilitate its uptake by the widest possible number of researchers, we developed `rflsgen`, an R package which allows to use the functionalities of `flsgen`. It can be built from sources using the GitHub repository, or directly downloaded from CRAN (https://cran.r-project.org/package=rflsgen).

Command-line interface (`https://github.com/dimitri-justeau/flsgen`): Finally, as part of the Java implementation, we developed a command-line interface (CLI) which offer access to most usages and parameters of `flsgen`. This CLI only requires Java Runtime Environment (JRE, version ≥ 8) installed, which makes it useful to launch large scale landscape generation on a remote computing server.

3 Use cases

3.1 Generating landscape series with fixed structure and varying spatial configurations

Neutral landscapes series are useful to assess the impact of landscape spatial configuration on ecological processes or to evaluate spatially-explicit models (e.g. fire spread simulation) with controlled datasets. However, for systematic analysis, it is necessary to ensure that landscape composition remains fixed while the spatial configuration is variable. In this use case, we illustrate how `flsgen` can be used to generate such landscape series by simulating patchy vegetation landscapes including three focal land-use classes: shrubland, savanna, and forest. The dimension of these landscapes is 500x500 pixels, with a resolution of 30x30 meters per pixel, which corresponds to a total extent of 22500 ha. First, we defined composition targets: PLAND = 20% for shrubland, 10% for savanna and forest; NP = 40 for shrubland, 30 for savanna, and 20 for forest, and AREA $\in$ [500, 3000] for shrubland, savanna, and forest. Then we generated a landscape structure satisfying these targets with `flsgen structure`. Maintaining this structure fixed, we generated a landscape series with a varying landscape configuration through the *terrain dependency* parameter (see Section 2.2) which varied from 0 to 1 with a step of 0.01, resulting in 101 landscapes. A continuous environmental gradient was generated on-the-fly by `flsgen` with the diamond-square algorithm and a roughness parameter of 0.2. A subset of the generated landscape is depicted in Figure 2. Finally, we evaluated the variation of spatial configuration in the landscape series through the *edge density* and *disjunct core area density* indices at the landscape level, using the `landscapemetrics` R package (Hesselbarth et al., 2019) (see Figure 2).

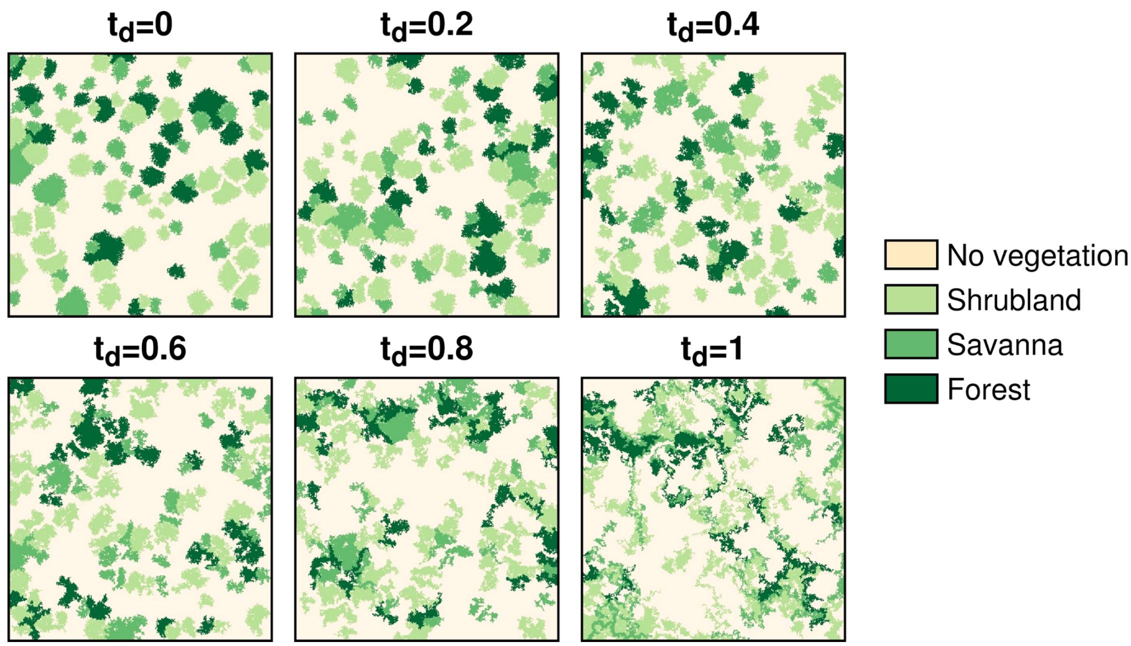

Figure 2: **(Use case 3.1)** Subset of the 101 generated 500x500 vegetation landscapes with fixed structure and varying spatial configuration.

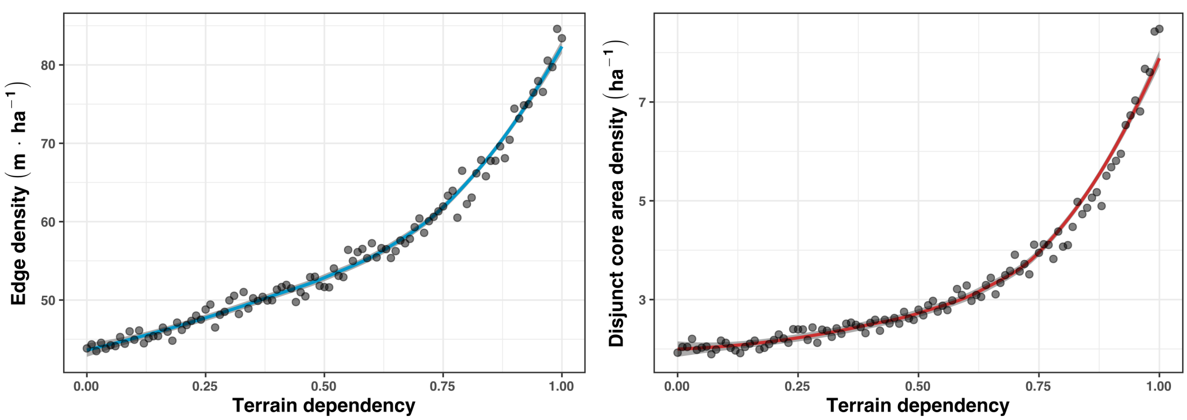

Figure 3: **(Use case 3.1)** Influence of the terrain dependency parameter (t_d) on landscape spatial configuration, measured with the edge density and the disjunct core area density indices.

3.2 Exploring correlations between fragmentation and connectivity patterns

Landscape fragmentation and connectivity pattern are known to impact ecological processes such as dispersal, gene flow, of fire resistance (Fahrig, 2003; Taylor et al., 1993). While the first refers to the structural patterns of habitat patches distribution, the second reflects the ability of species to migrate and disperse between habitat patches. Using the same scale as the previous use case (500x500 pixels at 30x30 meters resolution), we demonstrate how `flsgen` can be used to explore correlations between fragmentation and connectivity patterns, respectively measured with the *effective mesh size* (MESH, Jaeger, 2000), which was presented in the Introduction, and the *probability of connectivity* (PC Saura and Pascual-Hortal, 2007), which is a graph-based connectivity index based on a probabilistic connection model. Specifically, we generated a single focal class (e.g. rainforest) series of 2370 landscapes with MESH varying from 1000 pixels (90ha) $\pm 1\%$ to 60000 pixels (5400ha) $\pm 1\%$ with a step of 250 pixels (22.5ha). A subset of these landscapes is illustrated in Figure 4. For each MESH target, we left a high degree of freedom to other composition indices and generated 10 different landscape structures to ensure diversity in composition patterns. We computed the PC index for each generated landscape with the `Makurhini` R package, using the default probability threshold which is based on the inverse of the mean distance between patches (Godínez-Gómez and Correa Ayram, 2020). We plotted the relation between MESH and PC in the generated landscape series (see Figure 5), and evaluated the Pearson correlation coefficient ($r \approx 0.75$, p-value < 0.001), which suggests a strong positive linear correlation between MESH and PC. Given a value of MESH, we also observed a strict lower bound for PC corresponding to the case where the landscape is only composed of one patch. In this special case, PC equals MESH divided by the landscape area.

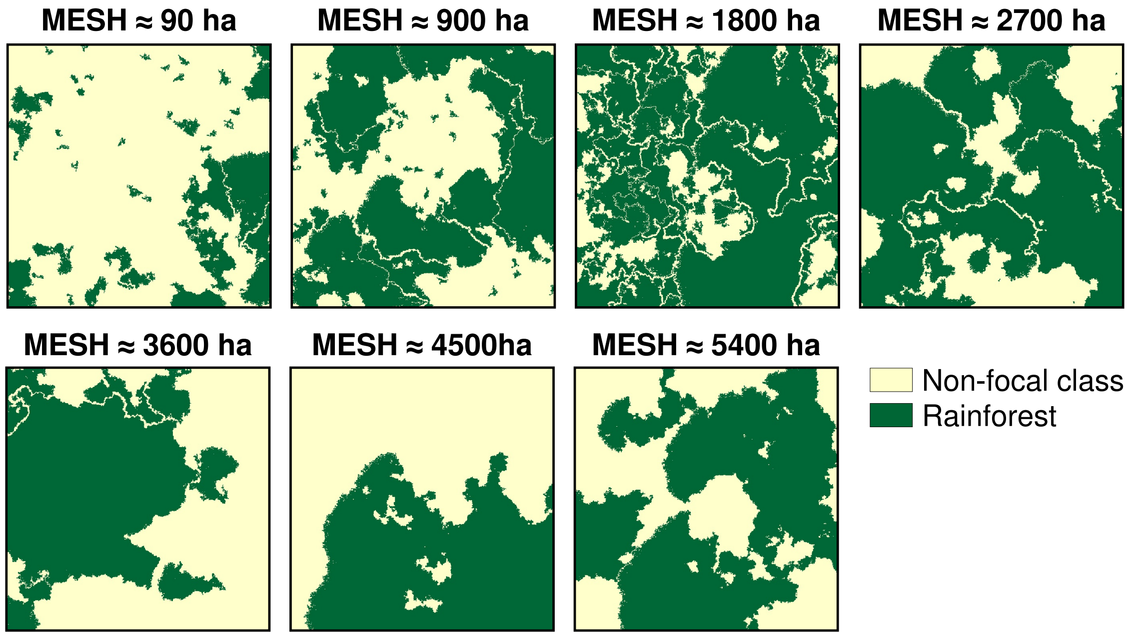

Figure 4: **(Use case 3.2)** Subset of the 2370 generated 500x500 landscapes with controlled effective mesh size (MESH).

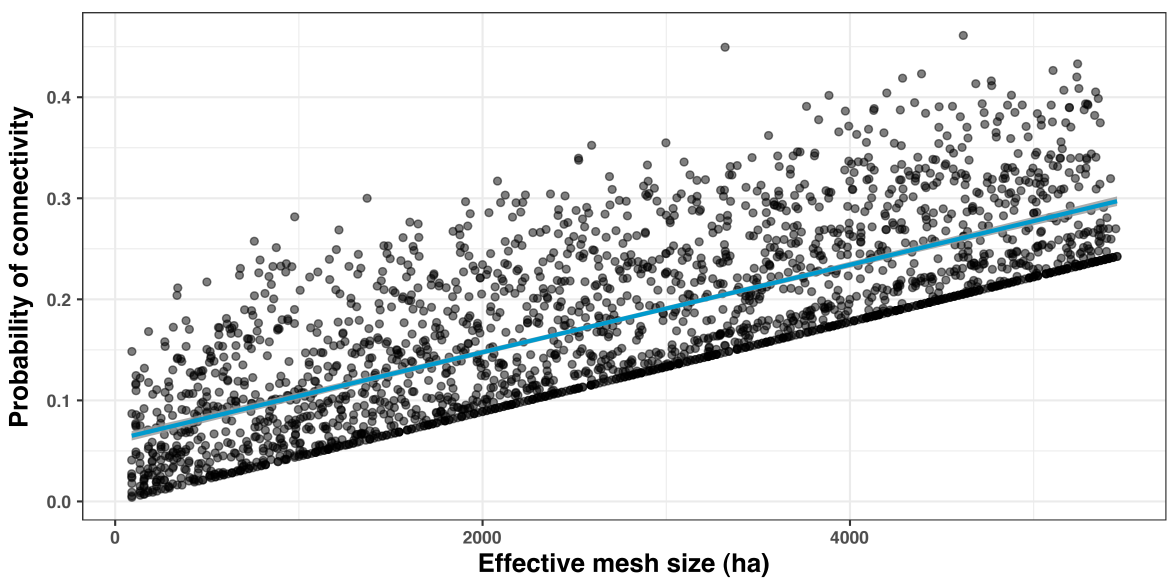

Figure 5: **(Use case 3.2)** Relation between the probability of connectivity (PC) index and the effective mesh size (MESH) evaluated from 2370 neutral landscapes of 500x500 pixels at 30x30 meters resolution (22500ha).

3.3 Recreating large landscape composition patterns

In this last use case, we illustrate how `flsgen` can be used to extract landscape structures from large real landscapes to recreate landscape composition patterns, with a focus on the forest cover of the main island of New Caledonia, which is a tropical archipelago in the South Pacific. First, we extracted 105x105 m New Caledonian forest cover data from the Copernicus Global Land Service database (Buchhorn et al., 2020), and produced a categorical raster map with two focal-classes: open and closed forest (see Figure 6). The dimension of the raster is 3297x2724, which corresponds to a total extent of 99,016 km^2, of which 16,030 km^2 are terrestrial. Then, we used `flsgen` to extract the landscape structure (with the 8-connectivity rule), which contains 13583 patches of open forest and 4906 patches of closed forest. Finally, we generated a neutral landscape using the New Caledonian digital elevation model as the continuous environmental gradient raster (see Figure 7).

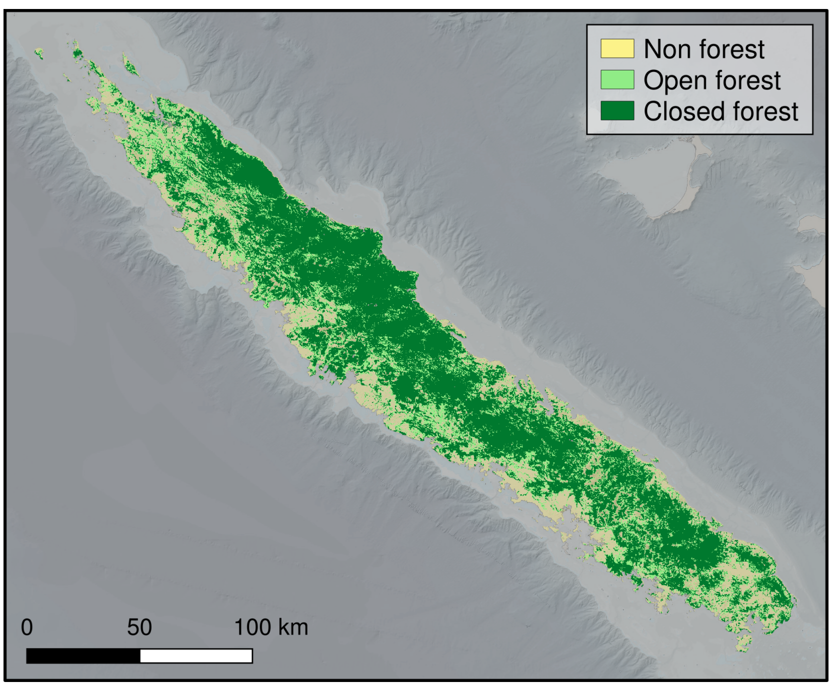

Figure 6: **(Use case 3.3)** Open and closed forest cover in the main island of New Caledonia, at 105x105 m resolution. Data from the Copernicus Global Land Service database.

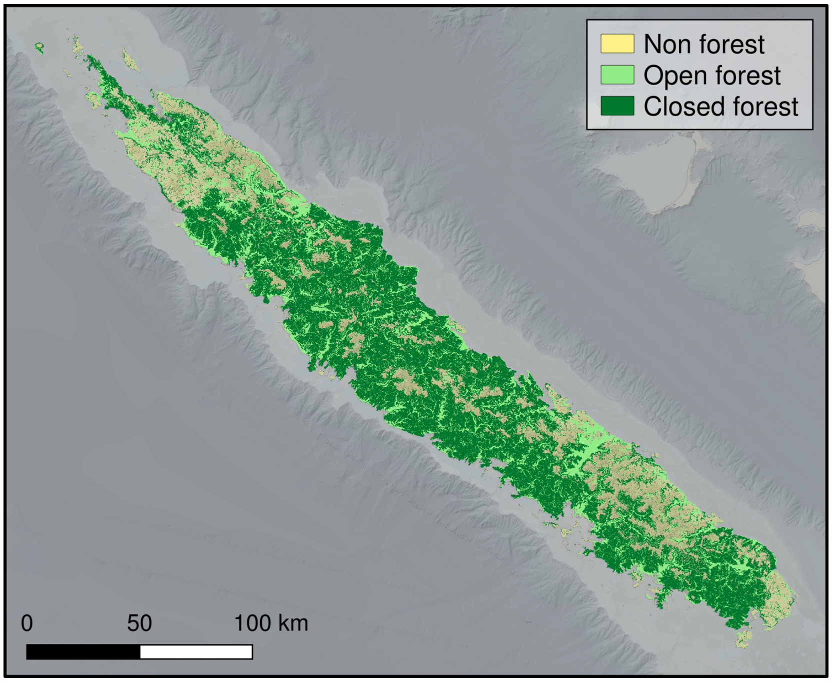

Figure 7: **(Use case 3.3)** Neutral landscape generated with `flsgen` recreating the landscape composition pattern of open and closed forest cover in the main island of New Caledonia (See Figure 6). The New Caledonian digital elevation model was used as the continuous environmental gradient in `flsgen`, with a terrain dependency set to 0.9. The 8-connectivity rule was used to extract the original landscape structure and to generate the neutral landscape.

4 Conclusion

In this article, we introduced `flsgen`, a neutral landscape generator that allows controlling many landscape composition and fragmentation indices. By separating the generation process into (i) a non-spatially-explicit constraint satisfaction phase and (ii) a spatially-explicit landscape generation phase, `flsgen` can generate large landscape series in small amounts of time (see Table 2). This new open-source software can support spatially explicit ecological simulations, evaluation of landscape indices or any other application that requires systematic and precise control of landscape composition and fragmentation indices. We aimed at making `flsgen` as accessible as possible through three available interfaces: a native Java API, an R package, and a command-line interface.

Use case	Number of landscapes	Landscape dimension	Number of focal classes	Total time
3.1	101	500x500	3	2.6 min
3.2	2370	500x500	1	3.6 h
3.3	1	3297x2724	2	54 s

Table 2: Use cases computation time (landscape generation).

Until now and to the best of our knowledge, *Landscape Generator* (`LG`, van Strien et al., 2016) was the only neutral landscape model allowing users to set target over landscape indices, although limited to low-resolution landscapes due to an exponentially increasing runtime. `flsgen` extends the possibilities offered by `LG` by implementing new landscape indices that can serve as targets and by allowing a fast generation of large landscapes, which opens new possibilities in terms of systematic experiments and analysis. Furthermore, the main difference between our approach and `LG` is that we focused on satisfying composition and fragmentation targets while controlling the spatial configuration with environmental gradients that can be produced by classical neutral models such as `NLMR` or `NLMpy` (Etherington et al., 2015; Sciaini et al., 2018). Consequently, `flsgen` is complementary to existing approaches: (i) classical neutral landscape models outputs can serve as continuous environmental gradients in `flsgen`, and (ii) landscape structures generated by `flsgen` can serve as preprocessed inputs in `LG`, whose targets are focused on spatial configuration indices. Although this second scenario is currently limited by `LG`'s computing time, we believe that our contribution can motivate further developments to overcome this limit and to provide more control over simulated data in ecological studies. In conclusion, by unlocking new possibilities for neutral landscape generation, we believe that `flsgen` is an asset to address novel questions in landscape ecology. In particular, we believe that it can support a better understanding of landscape indices behaviour and provide new insights to understand the relations between landscape patterns and ecological processes.

Acknowledgements

This work was funded by the ADMIRE project, a partnership between the IAC (New Caledonian Institute of Agronomy), the Cirad (French Agricultural Research Centre for International Development), and the North Province of New Caledonia.

Conflict of interest statement

The authors have no conflict of interest to declare.

Authors' contributions

All authors conceived the ideas and methodology. D.J. implemented the software and led the writing of the manuscript. All authors contributed critically to the draft, to software's documentation, testing, and gave final approval for publication.

Data availability statement

The software package and its source code is available on Zenodo at `https://doi.org/10.5281/zenodo.6386429` (Justeau-Allaire et al., 2022a) and `https://doi.org/10.5281/zenodo.6386420` (Justeau-Allaire et al., 2022b). It is also available on GitHub at `https://github.com/dimitri-justeau/flsgen` and `https://github.com/dimitri-justeau/rflsgen` (rflsgen). The R package `rflsgen` is also available on CRAN at `https://cran.r-project.org/package=rflsgen`.

References

J. Babí Almenar, A. Bolowich, T. Elliot, D. Geneletti, G. Sonnemann, and B. Rugani. Assessing habitat loss, fragmentation and ecological connectivity in Luxembourg to support spatial planning. Landscape and Urban Planning, 189:335–351, Sept. 2019. ISSN 0169-2046. doi: 10.1016/j.landurbplan.2019.05.004.

M. A. Bowers, S. F. Matter, J. L. Dooley, J. L. Dauten, and J. A. Simkins. Controlled experiments of habitat fragmentation: A simple computer simulation and a test using small mammals. Oecologia, 108(1):182–191, Oct. 1996. ISSN 1432-1939. doi: 10.1007/BF00333230.

M. Brand. Small polyomino packing. Information Processing Letters, 126:30–34, Oct. 2017. ISSN 0020-0190. doi: 10.1016/j.ipl.2017.06.004.

M. Buchhorn, B. Smets, L. Bertels, B. D. Roo, M. Lesiv, N.-E. Tsendbazar, M. Herold, and S. Fritz. Copernicus Global Land Service: Land Cover 100m: Collection 3: Epoch 2019: Globe, Sept. 2020.

E. C. B. Cambui, R. Nogueira de Vasconcelos, D. Boscolo, P. L. Bernardo da Rocha, and J. G. V. Miranda. GradientLand Software: A landscape change gradient generator. Ecological Informatics, 25:57–62, Jan. 2015. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2014.12.001.

R. J. Collins and G. W. Barrett. Effects of habitat fragmentation on meadow vole (Microtus pennsylvanicus) population dynamics in experiment landscape patches. Landscape Ecology, 12(2):63–76, Apr. 1997. ISSN 1572-9761. doi: 10.1007/BF02698208.

J. J. Cuervo and A. P. Møller. Demographic, ecological, and life-history traits associated with bird population response to landscape fragmentation in Europe. Landscape Ecology, 35(2):469–481, Feb. 2020. ISSN 1572-9761. doi: 10.1007/s10980-019-00959-9.

C. Dislich, E. Hettig, J. Salecker, J. Heinonen, J. Lay, K. M. Meyer, K. Wiegand, and S. Tarigan. Land-use change in oil palm dominated tropical landscapes—An agent-based model to explore ecological and socio-economic trade-offs. PLOS ONE, 13(1):e0190506, Jan. 2018. ISSN 1932-6203. doi: 10.1371/journal.pone.0190506.

T. R. Etherington, E. P. Holland, and D. O'Sullivan. NLMpy: A python software package for the creation of neutral landscape models within a general numerical framework. Methods in Ecology and Evolution, 6(2):164–168, 2015. ISSN 2041-210X. doi: 10.1111/2041-210X.12308.

L. Fahrig. Effects of Habitat Fragmentation on Biodiversity. Annual Review of Ecology, Evolution, and Systematics, 34(1):487–515, 2003. doi: 10.1146/annurev.ecolsys.34.011802.132419.

A. Fournier, D. Fussell, and L. Carpenter. Computer rendering of stochastic models. Communications of the ACM, 25(6):371–384, June 1982. ISSN 0001-0782. doi: 10.1145/358523.358553.

A. E. Frazier and P. Kedron. Landscape Metrics: Past Progress and Future Directions. Current Landscape Ecology Reports, 2(3):63–72, Sept. 2017. ISSN 2364-494X. doi: 10.1007/s40823-017-0026-0.

R. H. Gardner. RULE: Map Generation and a Spatial Analysis Program. In J. M. Klopatek and R. H. Gardner, editors, Landscape Ecological Analysis: Issues and Applications, pages 280–303. Springer, New York, NY, 1999. ISBN 978-1-4612-0529-6. doi: 10.1007/978-1-4612-0529-6_13.

C. Gaucherel, N. Giboire, V. Viaud, T. Houet, J. Baudry, and F. Burel. A domain-specific language for patchy landscape modelling: The Brittany agricultural mosaic as a case study. Ecological Modelling, 194(1):233–243, Mar. 2006. ISSN 0304-3800. doi: 10.1016/j.ecolmodel.2005.10.026.

O. Godínez-Gómez and C. A. Correa Ayram. Connectscape/Makurhini: Analyzing landscape connectivity (v1.0.0). Zenodo, Apr. 2020.

W. Hargrove, F. Hoffman, and P. Schwartz. A Fractal Landscape Realizer for Generating Synthetic Maps. Conservation Ecology, 6(1), Feb. 2002. ISSN 1195-5449. doi: 10.5751/ES-00371-060102.

M. H. K. Hesselbarth, M. Sciaini, K. A. With, K. Wiegand, and J. Nowosad. Landscapemetrics: An open-source R tool to calculate landscape metrics. Ecography, 42(10):1648–1657, 2019. ISSN 1600-0587. doi: 10.1111/ecog.04617.

T. Ibanez, V. Hequet, C. Chambrey, T. Jaffré, and P. Birnbaum. How does forest fragmentation affect tree communities? A critical case study in the biodiversity hotspot of New Caledonia. Landscape Ecology, 32(8):1671–1687, Aug. 2017. ISSN 0921-2973, 1572-9761. doi: 10.1007/s10980-017-0534-7.

J. A. Jaeger. Landscape division, splitting index, and effective mesh size: New measures of landscape fragmentation. Landscape Ecology, 15(2):115–130, Feb. 2000. ISSN 1572-9761. doi: 10.1023/A:1008129329289.

Justeau-Allaire, D., Blanchard, G., Ibanez, T. et al. Dimitri-justeau/flsgen: V1.1.0. Zenodo, 2022a. doi: https://doi.org/10.5281/zenodo.6386429.

Justeau-Allaire, D., Blanchard, G., Ibanez, T. et al. Dimitri-justeau/rflsgen: V1.0.0. Zenodo, 2022b. doi: https://doi.org/10.5281/zenodo.6386420.

K. McGarigal, S. A. Cushman, and E. Ene. FRAGSTATS v4: Spatial pattern analysis program for categorical and continuous maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. Available at the following web site: http://www. umass. edu/landeco/research/fragstats/fragstats. html, 2012.

M. C. Neel, K. McGarigal, and S. A. Cushman. Behavior of class-level landscape metrics across gradients of class aggregation and area. Landscape Ecology, 19(4):435–455, May 2004. ISSN 1572-9761. doi: 10.1023/B:LAND.0000030521.19856.cb.

G. Pe'er, G. A. Zurita, L. Schober, M. I. Bellocq, M. Strer, M. Müller, and S. Pütz. Simple Process-Based Simulators for Generating Spatial Patterns of Habitat Loss and Fragmentation: A Review and Introduction to the G-RaFFe Model. PLOS ONE, 8(5):e64968, May 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0064968.

C. Prud'homme, J.-G. Fages, and X. Lorca. Choco Documentation. 2017.

E. Rahimi, S. Barghjelveh, and P. Dong. Using the Lonsdorf model for estimating habitat loss and fragmentation effects on pollination service. Ecological Processes, 10(1):22, Mar. 2021. ISSN 2192-1709. doi: 10.1186/s13717-021-00291-8.

E. F. Rossi, P. van Beek, and T. Walsh. Handbook of Constraint Programming. page 969, 2006.

D. T. Rutledge. Landscape indices as measures of the effects of fragmentation: Can pattern reflect process? 2003.

S. Saura and L. Pascual-Hortal. A new habitat availability index to integrate connectivity in landscape conservation planning: Comparison with existing indices and application to a case study. Landscape and Urban Planning, 83(2):91–103, Nov. 2007. ISSN 0169-2046. doi: 10.1016/j.landurbplan.2007.03.005.

I. Schmiedel and H. Culmsee. The influence of landscape fragmentation, expressed by the 'Effective Mesh Size Index', on regional patterns of vascular plant species richness in Lower Saxony, Germany. Landscape and Urban Planning, 153:209–220, Sept. 2016. ISSN 0169-2046. doi: 10.1016/j.landurbplan.2016.01.012.

M. Sciaini, M. Fritsch, C. Scherer, and C. E. Simpkins. NLMR and landscapetools: An integrated environment for simulating and modifying neutral landscape models in R. Methods in Ecology and Evolution, 9(11):2240–2248, 2018. ISSN 2041-210X. doi: 10.1111/2041-210X.13076.

S. Seibold, C. Bässler, R. Brandl, L. Fahrig, B. Förster, M. Heurich, T. Hothorn, F. Scheipl, S. Thorn, and J. Müller. An experimental test of the habitat-amount hypothesis for saproxylic beetles in a forested region. Ecology, 98(6):1613–1622, 2017. ISSN 1939-9170. doi: 10.1002/ecy.1819.

B. S. Soares-Filho, G. Coutinho Cerqueira, and C. Lopes Pennachin. Dinamica—a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. Ecological Modelling, 154(3):217–235, Sept. 2002. ISSN 0304-3800. doi: 10.1016/S0304-3800(02)00059-5.

P. D. Taylor, L. Fahrig, K. Henein, and G. Merriam. Connectivity Is a Vital Element of Landscape Structure. Oikos, 68(3):571, Dec. 1993. ISSN 00301299. doi: 10.2307/3544927.

M. G. Turner. Landscape ecology: The effect of pattern on process. Annual review of ecology and systematics, 20(1):171–197, 1989.

M. van Strien, C. Slager, B. Vries, and A. Grêt-Regamey. An improved neutral landscape model for recreating real landscapes and generating landscape series for spatial ecological simulations. Ecology and Evolution, 6, May 2016. doi: 10.1002/ece3.2145.

T. Wiegand, E. Revilla, and K. A. Moloney. Effects of Habitat Loss and Fragmentation on Population Dynamics. Conservation Biology, 19(1):108–121, 2005. ISSN 1523-1739. doi: 10.1111/j.1523-1739.2005.00208.x.

K. A. With and A. R. Payne. An experimental test of the habitat amount hypothesis reveals little effect of habitat area but transient or indirect effects of fragmentation on local species richness. Landscape Ecology, July 2021. ISSN 1572-9761. doi: 10.1007/s10980-021-01289-5.