The Journal of Open Source Software

Omniscape.jl: Software to compute omnidirectional landscape connectivity

Vincent A. Landau[1], **Viral B. Shah**[2], **Ranjan Anantharaman**[3], **and Kimberly R. Hall**[4]

1 Conservation Science Partners, Inc., Fort Collins, Colorado, United States **2** Julia Computing Inc., Cambridge, Massachusetts, United States **3** Massachusetts Institute of Technology, Cambridge, Massachusetts, United States **4** The Nature Conservancy, Lansing, Michigan, United States

DOI: 10.21105/joss.02829

Software

- Review
- Repository
- Archive

Editor: Melissa Weber Mendonça

Reviewers:

- @juliohm
- @tpoisot

Submitted: 22 October 2020
Published: 29 January 2021

Summary

Omniscape.jl is a software package that implements the Omniscape algorithm (McRae et al., 2016) to compute landscape connectivity. It is written in the Julia programming language (Bezanson et al., 2017) to be fast, scalable, and easy-to-use. Circuitscape.jl (Anantharaman et al., 2020), the package on which Omniscape.jl builds and expands, abstracts landscapes as two-dimensional electrical networks and solves for current flow. The current flow that results represents landscape connectivity. Omniscape.jl is novel in that it produces maps of "omni-directional" connectivity, which provide a spatial representation of connectivity between every possible pair of start and endpoints in the landscape. These maps can be used by researchers and landscape managers to understand and predict how ecological processes (e.g., animal movement, disease transmission, gene flow, and fire behavior) are likely to manifest in geographic space. Omniscape.jl makes use of Julia's native multi-threading, making it readily scalable and deployable to high performance compute nodes. More information on the broader Circuitscape project, which is home to Circuitscape.jl and Omniscape.jl, can be found at circuitscape.org.

Motivation

Modeling where and how ecological processes are connected provides valuable information for landscape management and researchers. A common output from connectivity modeling efforts is a map that provides a spatial representation of connectivity by showing likely paths of flow for ecological processes. A common method for identifying flow or movement corridors uses graph-theory to identify least-cost paths (Bunn et al., 2000). The circuit-theoretic approach to connectivity modeling was a more recent innovation, and it has been gaining popularity over the past decade (Dickson et al., 2019; McRae, 2006; McRae et al., 2008). In the circuit-theoretic approach, the landscape is abstracted as a network of current sources, grounds, and resistors. The resulting current flow through the electrical network is then related to the movement or flow intensity of the ecological process of interest. These models were first implemented in the Circuitscape software package (Shah & McRae, 2008), which was recently updated and rewritten as Circuitscape.jl (Anantharaman et al., 2020) in the Julia programming language.

Circuitscape.jl is most often run in "pairwise" mode, where current flow is calculated between pairs of user-defined "cores," which are usually habitat patches. Results from this method can be highly sensitive to the location of cores. This can be problematic in cases where core location is arbitrary, or when there is uncertainty about where cores should be placed.

The Omniscape algorithm (McRae et al., 2016) offers an alternative, "coreless" approach to pairwise Circuitscape.jl and computes omni-directional landscape connectivity by implementing Circuitscape.jl iteratively in a moving window. By precluding the need to identify and delineate discrete cores, the Omniscape algorithm also allows for a more detailed evaluation of connectivity within natural areas that may otherwise be defined as cores in Circuitscape. Code for Python was developed to implement the Omniscape algorithm in McRae et al. (2016), but a user-friendly software package was not available. To fill this need, we developed Omniscape.jl, an easy-to-use software package written in Julia. Omniscape.jl is useful for modeling connectivity in landscapes that do not have discrete cores, for example landscapes that are a combination of natural, semi-natural, and human-modified lands, or in cases where understanding connectivity within natural areas is of interest.

The Omniscape Algorithm

Omniscape.jl works by applying Circuitscape.jl iteratively through the landscape in a circular moving window with a user-specified radius (Figure 1). Omniscape.jl requires two basic spatial data inputs: a resistance raster, and a source strength raster. The resistance raster defines the traversal cost (measured in ohms) for every pixel in the landscape, that is, the relative cost for the ecological process of interest to move through each pixel. The source strength raster defines for every pixel the relative amount of current (measured in amperes) to be injected into that pixel. In the case of modeling animal movement, a pixel with a high source strength corresponds to relatively more individuals originating from that pixel.

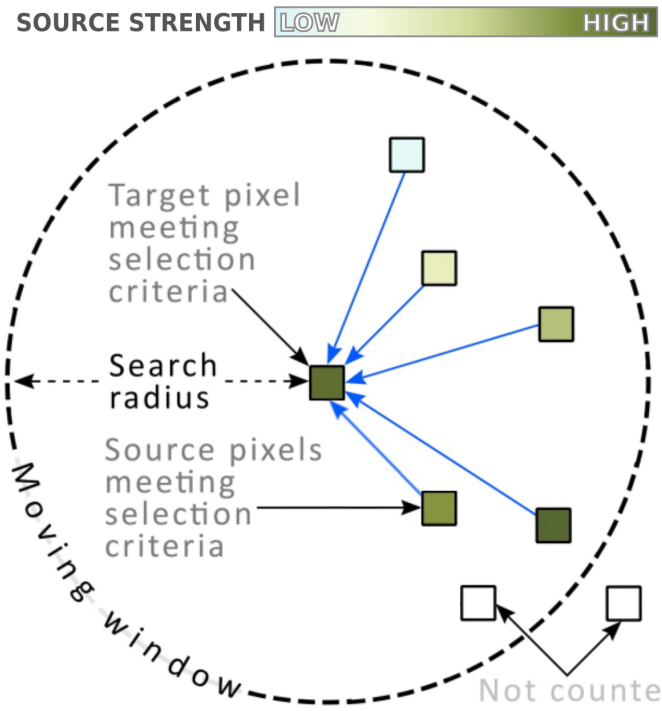

Figure 1: A diagram of the moving window used in Omniscape.jl, adapted with permission from McRae et al. (2016).

The algorithm works as follows:

1. The circular window centers on a pixel in the source strength surface that has a source

strength greater than 0 (or a user-specified threshold). This is referred to as the target pixel.
2. The source strength and resistance rasters are clipped to the circular window centered on the target pixel.
3. Every source strength pixel within the search radius that also has a source strength greater than 0 is identified. These are referred to as the source pixels.
4. Circuitscape.jl is run using the clipped resistance raster in "advanced" mode, where the target pixel is set to ground, and the source pixels are set as current sources. The total amount of current injected is equal to the source strength of the target pixel, and is divvied up among the source pixels in proportion to their source strengths.

Steps 1-4 are repeated for every potential target pixel. The resulting current maps from each moving window iteration are summed to get a final map of cumulative current flow. Individual moving window iterations can be run independently. Omniscape.jl makes use of Julia's multi-threaded parallel processing to solve individual moving windows in parallel.

In addition to cumulative current, Omniscape.jl also optionally provides two additional outputs: flow potential, and normalized cumulative current (Figure 2). Flow potential represents current flow under "null" resistance conditions and demonstrates what current flow would look like when unconstrained by resistance and barriers. Flow potential is calculated exactly as cumulative current flow, but with resistance set to one for the entire landscape. Normalized cumulative current flow is calculated by dividing cumulative current flow by flow potential. Normalized current helps identify areas where current is impeded or channelized (e.g., more or less current than expected under null resistance conditions). High values mean current flow is channelized, and low values mean current is impeded.

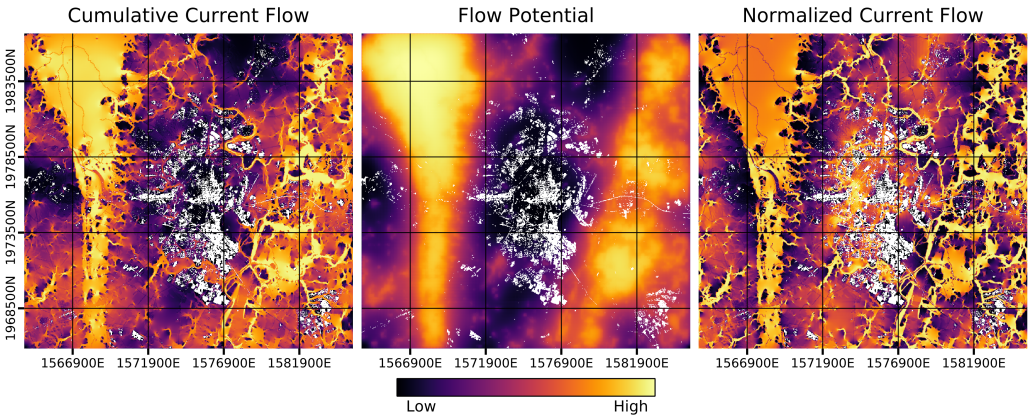

Figure 2: An example of the three different Omniscape.jl outputs. Outputs are from the Maryland forest connectivity example in the software documentation. Cumulative current flow shows the total current for each landscape pixel. Flow potential shows predicted current under resistance-free conditions. Normalized current shows the degree to which a pixel has more or less current than expected under resistance-free conditions (cumulative current flow divided by flow potential). Each layer is visualized using a quantile stretch with 30 breaks. Axes show easting and northing coordinates for reference.

Usage

Omniscape.jl is run from the Julia REPL. It offers a single user-facing function, `run_omniscape`, which has two methods. The first method accepts a single argument specifying the path to an INI file that contains input data file paths and run options. Spatial data inputs can be in either ASCII or GeoTIFF raster formats, and outputs can also be written in either format. The

second method of `run_omniscape` accepts arrays representing resistance and other spatial data inputs, and a dictionary of arguments specifying algorithm options. A complete user guide for Omniscape.jl, including installation instructions, function documentation, examples, and a complete list of options and their defaults, can be found in the Omniscape.jl documentation.

Acknowledgments

Development of this software package was made possible by funding from NASA's Ecological Forecasting program (grant NNX17AF58G) and the Wilburforce Foundation. This software package would not have been possible without Brad McRae (1966-2017), the visionary behind Circuitscape, the Omniscape algorithm, and several other software tools for assessing connectivity. Aaron Jones developed the diagram in Figure 1. Aaron Jones, Carrie Schloss, Melissa Clark, Jim Platt, and early Omniscape.jl users helped steer software development by providing valuable feedback and insight.

References

Anantharaman, R., Hall, K., Shah, V. B., & Edelman, A. (2020). Circuitscape in Julia: High performance connectivity modelling to support conservation decisions. *JuliaCon Proceedings*, *1*(1), 58. https://doi.org/10.21105/jcon.00058

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bunn, A. G., Urban, D. L., & Keitt, T. H. (2000). Landscape connectivity: A conservation application of graph theory. *Journal of Environmental Management*, *59*(4), 265–278. https://doi.org/10.1006/jema.2000.0373

Dickson, B. G., Albano, C. M., Anantharaman, R., Beier, P., Fargione, J., Graves, T. A., Gray, M. E., Hall, K. R., Lawler, J. J., Leonard, P. B., & others. (2019). Circuit-theory applications to connectivity science and conservation. *Conservation Biology*, *33*(2), 239–249. https://doi.org/10.1111/cobi.13230

McRae, B. H. (2006). Isolation by resistance. *Evolution*, *60*(8), 1551–1561. https://doi.org/10.1111/j.0014-3820.2006.tb00500.x

McRae, B. H., Dickson, B. G., Keitt, T. H., & Shah, V. B. (2008). Using circuit theory to model connectivity in ecology, evolution, and conservation. *Ecology*, *89*(10), 2712–2724. https://doi.org/10.1890/07-1861.1

McRae, B. H., Popper, K., Jones, A., Schindel, M., Buttrick, S., Hall, K. R., Unnasch, R. S., & Platt, J. (2016). Conserving nature's stage: Mapping omnidirectional connectivity for resilient terrestrial landscapes in the pacific northwest. *The Nature Conservancy, Portland, Oregon*. https://doi.org/10.13140/RG.2.1.4158.6166

Shah, V. B., & McRae, B. H. (2008). Circuitscape: A tool for landscape ecology. *Proceedings of the 7th Python in Science Conference*, *7*, 62–66.