


CAPS traversability metric in R

August 23, 2021, 2021

Bradley W. Compton
Landscape Ecology Program
University of Massachusetts
Amherst, MA 01003 USA
bcompton@eco.umass.edu

The CAPS (Conservation Assessment and Prioritization System) traversability metric (also known as old-style connectedness and local connectedness) builds a resistant kernel for each focal cell in landcover using resistances supplied in a table. In normal mode, each focal cell gets the sum of the volume of a resistant kernel built on that cell, divided by the sum of a standard kernel built on that cell. This gives a measure of local connectedness for each focal (usually undeveloped) cell in the landscape. Alternatively, example points may be specified, and resistant kernels are built for each example cell; these kernels are summed where they overlap.

 This document describes installing **traverse** package, and gives background and examples. See **traverse.pdf** for details on running traverse in R.

traverse runs in R v. 4.x and later (it may work in earlier versions, but hasn't been tested). I suggest you run from within RStudio. The free version is just fine. You'll probably want to run in 64-bit R so you're not stuck with toy examples.

- R (<https://www.r-project.org/>)
- RStudio (<https://www.rstudio.com/>)

Package and documentation

http://landeco.umass.edu/web/traverse/traverse_2.0.0.zip. The most recent version of the traverse package. In RStudio, use Tools > Install packages... > Install from Package Archive File. Use `library(traverse)` to load it. At that point you can type `help(traverse)` for help, and `traverse(...)` to run it. You can start by running the examples from the help.

<http://landeco.umass.edu/web/traverse/docs.zip> contains:

- **CAPS Traversability metric in R.PDF**. The most recent version of this document.
- **traverse.pdf**. Complete traverse documentation. This is a PDF version of `help(traverse)` from R.
- **ResistantKernelAlgorithm.pdf**. Documents the details of the resistant kernel algorithm.

Required packages

- **gridprocess**. In RStudio, use Tools > Install packages... > Install from Package Archive File.

For R v. 4.x.x:

http://landeco.umass.edu/web/code/r4.0.4/gridprocess_0.1.4.zip

For R 3.6.x:

http://landeco.umass.edu/web/code/r3.6.1/gridprocess_0.1.4.zip

- **rgdal, raster, spatial, rgeos, utils, stats** from CRAN. You can install these from within RStudio with Tools > Install packages... > Repository (CRAN).

Data to run the examples in the documentation are in the traverse package, under `traverse\inst\exampledata\`.

Release notes

v 2.0.0 August 23, 2021 Rewrite from old version that required ESRI grid API

Discussion

The bandwidth selects the size of the kernels, thus the degree of smoothing over the landscape. Small bandwidths (a few hundred meters) are appropriate for getting local detail, and make sense for focal species with small homeranges and limited dispersal. Larger bandwidths (up to several kilometers) make sense to far-ranging focal species and large-scale landscape processes. For Massachusetts CAPS, we used *bandwidth* = 2000; for the Resilient Sites project, TNC used *bandwidth* = 3000. There's no right answer; experiment until you get something that looks good to you. Note that runtime is proportional to the square of (*bandwidth* × *search*). You can set *search* to a smaller number to reduce runtime, at the cost of losing the tail edges of kernels.

Resistance values represent the cost to an organism of moving through a particular cover type in terms of physiological cost, willingness to cross, and mortality. Values range from 1.0 (minimal cost, usually assigned to an animal's habitat or natural cover) up to large values representing absolute barriers. Resistance is simply a multiplier on distance, giving functional distance; for example, moving through one cell with a resistance of 5 has the same cost as moving through five cells with resistance of 1. In this version of traversability, resistances are constant for each landcover type, as opposed to the full CAPS version, in which resistance is derived from ecological distance in settings space. This simpler version has minimal data requirements (a landcover map and resistance values for each type), but is more limited in representing varying resistances of natural landcover types.

You can specify the focal cells to run for, typically natural cover, though you might want to run just for forest, or just wetlands, for instance. Each focal cell will get a result; non-focal cells get a value of 0. By default, any cell with a resistance of 1.0 is treated as a focal cell, but you may supply a higher resistance cutoff with *focal*, or use *focal* to specify

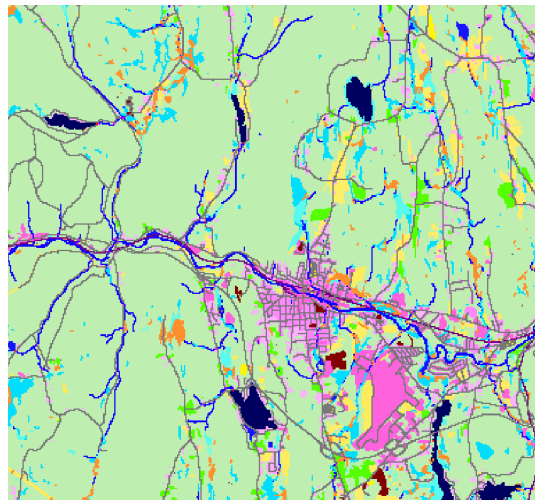
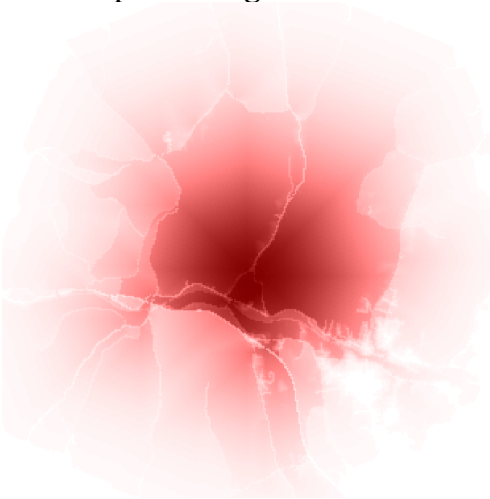
a column in the resistance table to indicate whether each landcover type is treated as a focal cell.

Normally, when building a kernel for a focal cell, the resistance of the focal cell is ignored. You can use `focalresist = TRUE` to count the focal cell's resistance.

This version is designed to run for relatively small landscapes, up to a several thousand km², depending on how patient you are. It is single-threaded, with no provisions for file locking. Traversability is computationally intensive, so running for large landscapes (such as states or provinces) may require parallel processing on many machines.

You can specify a window within a larger grid to run for a small area of a large landscape. If your landscape grid is large (a state, province, or country), you'll probably need to use the *window* option to run for a smaller area (even if you're using *example* =). Specify a polygon shapefile. `traverse()` will read a buffer beyond the window you specified so that you won't have problems with edge effects. If your window is at the edge of the data, `traverse()` will do edge correction by assuming that whatever is beyond the data edge is essentially the same as data within the local window.

You can use the *example* option to build kernels for individual cells. This can be helpful for selecting a bandwidth and resistances. When *example* is used, the complete kernels are returned in the result grid, rather than having their volume summed. Individual kernels are fun to look at. They're also fast to run. Here's a single kernel and the landscape it was generated for:



Here's are a couple of example calls using the sample data. The first builds three example kernels, and the second runs the traversability metric for every undeveloped cell in the window (using a smaller bandwidth).

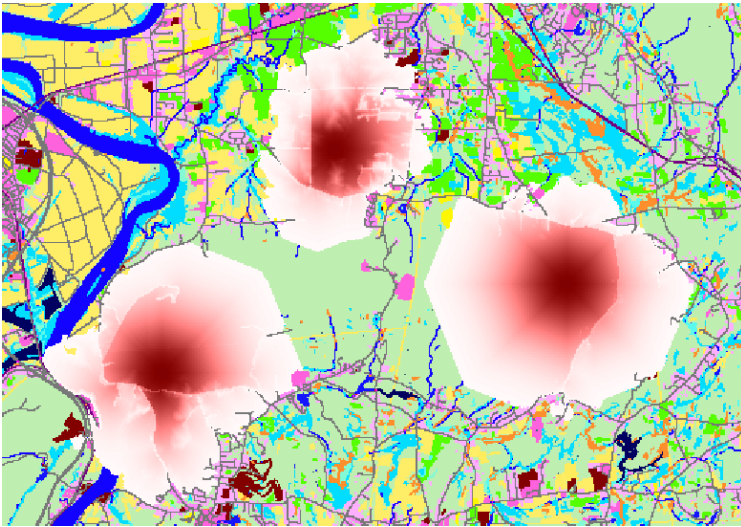
```
> traverse(1000, resist = 'd:/caps/grids/resist.txt', land = 'd:/caps/grids/land',
  result = 'd:/caps/results/test22', window = 'd:/caps/grids/clip',
  example = c(110377, 893424, 114009, 898109, 118869, 895284))
```

Running traversability for 3 example cells...

100%

Traversability has been run for d:/caps/grids/land; results written to d:/caps/results/test22

>



```
> traverse(500, resist='d:/caps/grids/resist.txt', land = 'd:/caps/grids/land',
  result = 'd:/caps/results/test24', window = 'd:/caps/grids/clip')
```

Running traversability for 149,517 cells...

100%

Traversability has been run for d:/caps/grids/land; results written to d:/caps/results/test24

>

