Generating landscapes with the `rlandsacpe` package

Gregor Passolt

September 7, 2012

Contents

The `rlandscape` package is intended to make it easy to simulate random landscapes for testing harvest scheduling models. This vignette will cover use of the two primary functions, `rlandscape` and `rland`, and is aimed at users that are new to R. Most users will probably find `rland` the more useful, but should have a basic understanding of what `rlandscape` does so they know what they're getting. To learn about how `rlandcsape` works, please refer to Gregor Passolt, Miranda J. Fix, and Sandor F. Toth (in review). A Voronoi Tessellation-based Approach to Generate Hypothetical Forest Landscapes, Canadian Journal of Forest Research.

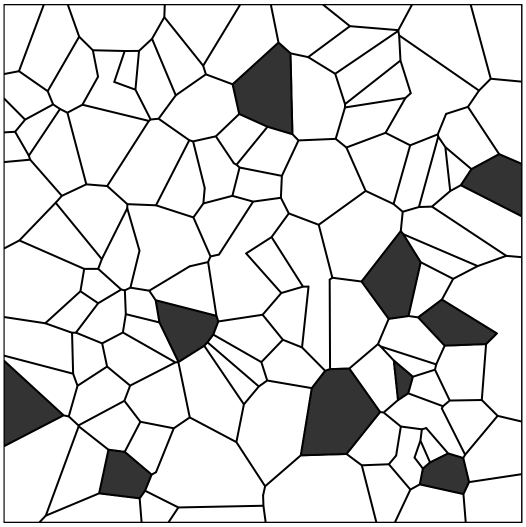
1 Generating a single landscape with `rlandscape`

The foundation of the package is `rlandscape`, which generates a single landscape at a time. Its output (a landscape object) can be assigned to a variable for analysis or saving. The first step is to download the `rlandscape` package. If you're reading this vignette, that's probably already done, but just in case the command is `install.packages("rlandscape")`. Other dependencies will be installed automatically, which may take a minute. This only needs to be performed once. Every time you start a new R session, you will need to load the package, which can be done either with the `require` or `library` commands.

```r
library(rlandscape)  ## loading the package
```

Now that the package is loaded, all the associated functions are available. Let's get started creating a landscape

```r
myLand <- rlandscape()
plot(myLand)
```



```
myLand$stats

##   nOut degMean degSD areaCV hAsp  n1 n2 n3 n4 pMerge pHole
## 1  100    4.86 1.511  55.15    1 100  0  0  0    0.1   0.1
```

Particular features of this landscape can be extracted using the $ operator. The `stats` display the final number of polygons (nOut), the mean of the degree distribution (degMean), the standard deviation of the degree distribution (degSD), the coefficient of variation of the area distribution (areaCV), the horizontal:vertical aspect ratio (hAsp), the number of points placed by each of the 4 point placement methods (uniform, lattice, cluster, inhibition, respectively), the proportion of edges deleted to merge two polygons together (pMerge), and the proportion of polygons deleted to become holes (pHole). The adjacency table is available as `adj`, and the areas are in the `dir.area` column of a summary table that includes information for all of original points, such as their x-y coordinates, whether they were deleted, and whether they were merged with another polygon. Note that, due to the deletions and merge events, the polygon numbering will not be sequential. For example, if polygon 2 is deleted, another polygon will not be renamed “2”.

```
head(myLand$adj)

## [[1]]
## [1]  13  20  67 103
##
## [[2]]
## [1]  16  32  39  42  62  69 101
##
## [[3]]
## [1]  51  89 103 115
##
## [[4]]
## [1] 21 30 46 49 83
##
## [[5]]
## [1]  6 15 49 83
```

```
##
## [[6]]
## [1]  5 15 23

head(myLand$summary)

##           x      y n.tri del.area  del.wts n.tside nbpt dir.area  dir.wts
## [1,] 0.6262 0.9559     6 0.008947 0.010097       5    2 0.010539 0.010539
## [2,] 0.2227 0.9439     9 0.008992 0.010147       7    2 0.013722 0.013722
## [3,] 0.7180 0.8291     5 0.004188 0.004727       5    0 0.005055 0.005055
## [4,] 0.5592 0.3903     6 0.005205 0.005874       6    0 0.006399 0.006399
## [5,] 0.4710 0.2936     5 0.003600 0.004062       5    0 0.003873 0.003873
## [6,] 0.4586 0.2792     4 0.002910 0.003285       4    0 0.003542 0.003542
##      holeThese mergedWith
## [1,]         0          0
## [2,]         0          0
## [3,]         0          0
## [4,]         0          0
## [5,]         0          0
## [6,]         0          0
```

You can give arguments to `rlandcsape` to alter the landscape under construction. For example, a landscape of 500 polygons, with a 2:1 aspect ratio and a very patchy composition would be given by:

```
land2 <- rlandscape(n = 500, hAsp = 2, pHole = 0.5)
```

You can access the helpfile for `rlandscape` by entering `?rlandscape` at the console. It describes all of the options available as well as the defaults.

2 Batch generation using `rland`

The `rland` function is used when, rather than specifying control parameters for the point processes used to generate landscapes, you want to specify the characteristics of the resulting landscape. It also makes it easy to generate and save any number of landscapes.

The most workhorse arguments to `rland` are `targets` and `bounds`. The `targets` sets the range of landscape characteristics the algorithm will "aim" for, while `bounds` sets the range of landscape characteristics that will be accepted. The target ranges must fall inside the bounds ranges. See the `rland` help file (enter `?rland`) for information on defaults.

`Rland` is designed to write its output to files rather than display it in R. When you start an R session, there is a "working directory" to which, by default, R will save files (or look for files to load). You can see or change the current working directory with the commands `getwd` and `setwd`, or through file menus which depend on your operating system and interface. The working directory is where `rland` will write its output unless you specify a full filepath. The `filename` argument to `rland` is for the name of the run. For example, if you use `filename = "land-sim"` and create 10 landscapes, they will be named and saved as `land-sim-01` to `land-sim-10` in the working directory. To have them saved somewhere else, you could set `filename = "C:/otherDirectory/land-sim"`. The default filename is `"landscape"`.

2.1 Examples

So, if you wanted to generate 50 landscapes with between 200 and 300 polygons, and save the plots of each landscape, this would do it:

```
    rland(targets = list(n = c(200, 300)),
          bounds = list(n = c(200, 300)),
          reps = 50,
          filename = "run1",
          savePlot = TRUE)
```

A similar run with more specifications would look like this:

```
rland(targets = list(n = c(200, 300),
                     degMean = c(4.8, 5.2),
                     areaCV = c(50, 70),
                     hAsp = c(1, 5)),
      bounds = list(n = c(175, 325),
                    degMean = c(4.8, 5.2),
                    areaCV = c(45, 75)),
      reps = 50, filename = "run2", savePlot = TRUE)
```

By default, the plots are not saved, but the adjacencies and areas are, as well as a summary table giving descriptive statistics about each of the landscapes. This can be viewed in R by

```
run1.summary <- read.table("run1_summary.csv")
head(run1.summary)
hist(run1.summary$degMean) ## plotting a histogram of the degree means
```

A assortment of 100 landscapes with random characteristics is created by

```
rland(reps = 100, method = "random", filename = "random_landscape")
```

2.2 GUI

A graphical user interface (GUI) is available to interact with `rland`. All the arguments of `rland` can be accessed through the GUI. To start the GUI, simply enter `rland.gui()` in the R console.