Chapter 7
What Constitutes a Significant Difference in Landscape Pattern?

Tarmo K. Remmel and Marie-Josée Fortin

OBJECTIVES

Characterizing landscape spatial structure can provide insights about the underlying mechanisms that generate pattern. Quantifying spatial structure enables analysis of landscape change over time as well as comparisons among different locations. Although numerous landscape metrics (LMs) exist to quantify spatial structure and characterize a landscape, how do we know when two landscapes significantly differ? As a single landscape represents only one replicate, its metrics are not statistics; thus, testing for differences between two landscapes becomes difficult. To address this problem, randomization procedures can help assess statistical significance using simulation approaches that assess whether the observed spatial structure could have occurred by chance alone. In this chapter, exercises will allow students to accomplish the following objectives:

1. Perform significance tests of landscape metrics based on a randomization procedure using a simulation model;
2. For one landscape, assess whether LM values are significantly different than those from landscapes of similar class proportions and spatial autocorrelation;
3. Determine where an LM value falls within its potential distribution, after controlling for class proportions and spatial autocorrelation;
4. Evaluate the statistical differences of LM values from two landscapes; and
5. Graphically produce, present, and discuss results in an R software environment.

Exercise 1 uses a simple, simulated binary landscape to explore null hypothesis testing, whereas Exercise 2 addresses statistical significance for more than one

T.K. Remmel (✉)
York University, Toronto, ON, Canada
e-mail: remmelt@yorku.ca

M.-J. Fortin
University of Toronto, Toronto, ON, Canada

S.E. Gergel, M.G. Turner (eds.), *Learning Landscape Ecology*,
DOI 10.1007/978-1-4939-6374-4_7

landscape. In Exercise 3, an applied example is explored within the context of making landscape restoration decisions. All of our implementation, analysis, graphics, and exercises are produced within the R statistical software environment (R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org/, 2010) and students are provided with all coding and files needed (from the book website). This lab requires installation of R (or R Studio) as well as loading the following libraries (*SDMTools, fields*). Some basic familiarity with R is helpful for completing this lab, and knowledge of concepts from Chapter 5 is assumed. Where relevant, advice is provided for adapting the code to the user's own research.

INTRODUCTION

Landscape metrics describe and quantify the spatial pattern of categorical landscape data (McGarigal 2002, as well as Chapters 4 and 10). LMs are often used to compare spatial patterns of landscapes from distinct regions or within the same landscape through time. However, such comparisons based on LM are valid however, only when the proportions, p_i, of each category, *i,* are the same between the landscapes (Turner et al. 2001). The reason for comparing only landscapes with similar values of p is because of the strong relationship between landscape metrics and the proportion of the landscape occupied by that cover type.

Another key factor affecting spatial pattern is spatial autocorrelation (i.e., values at nearby locations are more similar than by chance; Fortin and Dale 2005) which you explored in Chapter 5. Clumped, aggregate patterns occur when correlations are positive whereas dispersed, disaggregate patterns occur when correlations are negative. Spatial autocorrelation is also confounded with class proportions. As a result, in comparisons among landscapes of different proportions, one cannot often determine whether the differences are due to class proportions, spatial autocorrelation, or some other process (e.g., fragmentation).

Prior to comparing LM among landscapes, one should first determine whether the observed LM values for a single landscape could have occurred by chance. This is not a trivial task because probability distributions of LM are largely unknown, and cannot be analytically derived, especially when both proportion and spatial autocorrelation need to be accounted for (Fortin et al. 2003; Remmel and Csillag 2003). As a result, the distributions of LM must be constructed empirically using randomization and simulation procedures. The lack of analytically derived distributions in ecological studies often requires computer-intensive randomization procedures, such as resampling, Monte Carlo methods, or bootstrapping (Manly 2006; Fortin et al. 2012), in order to perform significance testing.

While randomization procedures are very flexible methods, they are usually based on the assumption of data independence. In a spatial context, this independence assumption corresponds to a complete spatial randomization (CSR), where the values of a variable are equally likely to be distributed over the entire area. In the presence of spatial autocorrelation, this assumption is unlikely to be valid, and hence a null distribution should be generated using stochastic simulation approaches. Stochastic simulations can produce highly replicated landscapes with known levels of spatial autocorrelation and class proportions in order to generate the empirical distributions needed for significance testing (Fortin et al. 2003; Remmel and Csillag 2003). Once you complete the lab, we recommend you re-read this introductory material to help solidify these concepts and new terminology.

For simplicity, the examples we use in this lab are binary landscapes, and their proportions and spatial structure affect the values of the LM (Proulx and Fahrig 2010). Figure 7.1 (after Remmel and Csillag 2003) illustrates the dependence

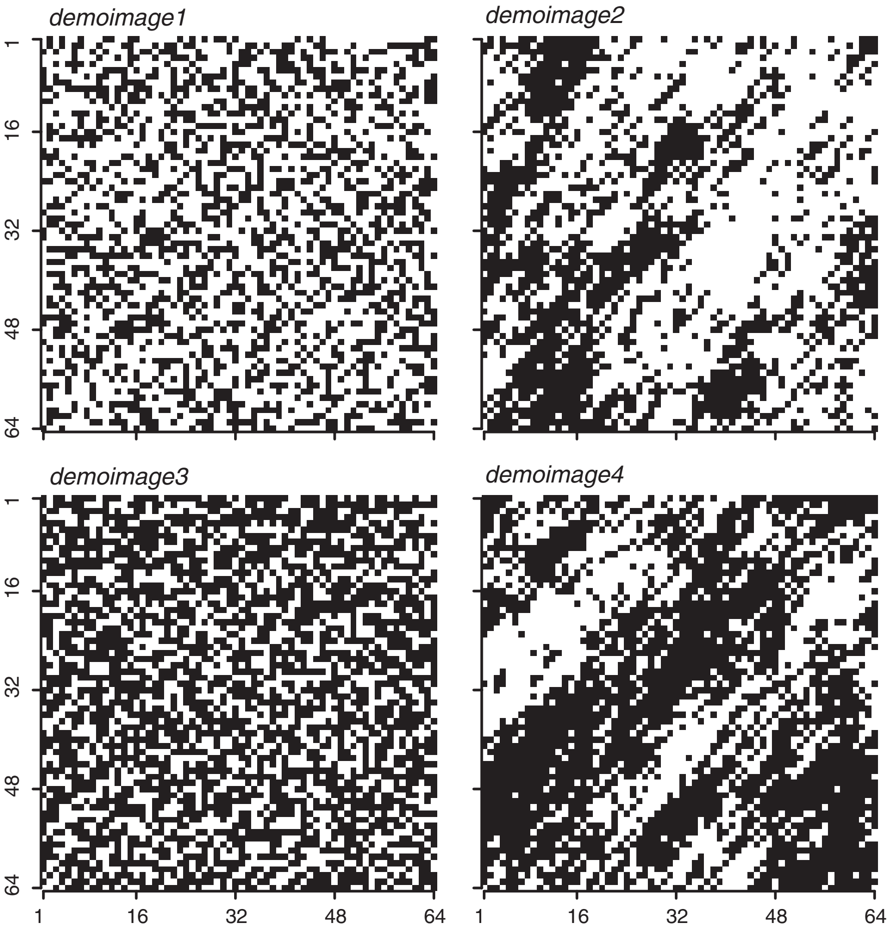

Figure 7.1 Four simulated landscapes (64×64 pixels) with specified class proportions and degrees of spatial autocorrelation. Landscapes have the following class-level proportions: in the *top row* Black = 0.40 and White = 0.60; while in the *bottom row* Black = 0.60 and White = 0.40. The levels of spatial autocorrelation are fixed: random ($\rho_W = 0.00$) in the left column and distinct patches ($\rho_W = 0.49$) in the right column

between p and spatial autocorrelation: when the proportion increases, the degree of spatial autocorrelation increases as well; therefore, LM should be compared between landscapes with similar proportions occupied by the category of interest.

Simulated Landscapes: A Stationary Stochastic Random Field Simulator

Several algorithms exist to generate binary landscapes with a known degree of spatial autocorrelation (Hargrove et al. 2002; Fortin et al. 2003; Remmel and Csillag 2003; Gardner and Urban 2007; as well as Chapter 6 in this book) but none exist (with the same degree of control) for landscapes with multiple categories, due to the inherent complexity of spatial patterns possible. Hence, the exercises in this chapter focus only on binary landscapes and significance testing of class-level metrics.

We implement **conditional autoregressive (CAR) simulations** (Cressie 1993) for generating null distributions of **binary** (two land cover classes: black and white), **isotropic** (the spatial autocorrelation parameters do not vary with direction), **stationary** (the spatial autocorrelation parameters are constant across the entire map) landscapes that rely on the stochastic random field model (Remmel and Csillag 2003). The CAR model utilizes a covariance matrix, **C**:

$$\mathbf{C} = \left(\mathbf{I} - \rho\mathbf{W}\right)^{-1}$$

where **I** is an **identity matrix** (a matrix of all zeros except for the diagonal that is filled with ones), and **W** is a **connectivity matrix** that defines which cells are considered neighbors (nearest neighbors = 1) or not (=0). Also, $\boldsymbol{\rho}$ is the **spatial autocorrelation** parameter, similar to Moran's *I* (Moran 1950) that ranges from −1 (repulsion) to 1 (attraction). Our implementation ranges from complete spatial randomness ($\rho_W = 0.00$) to highly spatially autocorrelated, where ρ approaches 1. For continuous data, ρ can be estimated using Whittle's algorithm (Whittle 1954) which is explained in more detail in Appendix A. As this correction factor requires intensive computation, it has already been performed and stored as a lookup table in the provided Remmel–Fortin code as object *DIFF50* and is used internally when estimating ρ_W.

Landscape Metrics

The LM computed in the provided Remmel–Fortin code are class-level metrics (Table 7.1) from FRAGSTATS (McGarigal and Marks 1995) and implemented within the *SDMTools* (VanDerWal et al. 2011) library in R. Here, numbers 1–38 to refer to specific class metrics corresponding to the black category, with 39–76 for white as computed by *SDMTools* (Table 7.1). All LMs, where required, are computed using nearest neighbors (4-neighbor rule).

Table 7.1 Explanation of 38 class-level landscape metrics

Low	High	Metric acronym	Definition of class-level metric
1	39	`class`	Particular patch type from the original input data
2	40	`n.patches`	Number of patches of a particular patch type or in a class
3	41	`total.area`	Sum of the areas (m^2) of all patches of the corresponding patch type
4	42	`prop.landscape`	Proportion of the total landscape represented by this class
5	43	`patch.density`	Numbers of patches of corresponding patch type divided by total landscape area (m^2)
6	44	`total.edge`	Total edge length of a particular patch type
7	45	`edge.density`	Edge length on a per unit area basis that facilitates comparison among landscapes of varying size
8	46	`landscape.shape.index`	A standardized measure of total edge or edge density that adjusts for the size of the landscape
9	47	`largest.patch.index`	Largest patch index quantifies the percentage of total landscape area comprised by the largest patch
10	48	`mean.patch.area`	Average area of patches
11	49	`sd.patch.area`	Standard deviation of patch areas
12	50	`min.patch.area`	Minimum patch area of the total patch areas
13	51	`max.patch.area`	Maximum patch area of the total patch areas
14	52	`perimeter.area.frac.dim`	Perimeter-area fractal dimension equals 2 divided by the slope of regression line obtained by regressing the logarithm of patch area (m^2) against the logarithm of patch perimeter (m)
15	53	`mean.perim.area.ratio`	Mean of the ratio patch perimeter. The perimeter-area ratio is equal to the ratio of the patch perimeter (m) to area (m^2)
16	54	`sd.perim.area.ratio`	Standard deviation of the ratio patch perimeter
17	55	`min.perim.area.ratio`	Minimum perimeter area ratio
18	56	`max.perim.area.ratio`	Maximum perimeter area ratio
19	57	`mean.shape.index`	Mean of shape index
20	58	`sd.shape.index`	Standard deviation of shape index
21	59	`min.shape.index`	Minimum shape index
22	60	`max.shape.index`	Maximum shape index
23	61	`mean.frac.dim.index`	Mean of fractal dimension index
24	62	`sd.frac.dim.index`	Standard deviation of fractal dimension index
25	63	`min.frac.dim.index`	Minimum fractal dimension index
26	64	`max.frac.dim.index`	Maximum fractal dimension index
27	65	`total.core.area`	Sum of the core areas of the patches (m^2)
28	66	`prop.landscape.core`	Proportional landscape core
29	67	`mean.patch.core.area`	Mean patch core area
30	68	`sd.patch.core.area`	Standard deviation of patch core area

(continued)

Table 7.1 (continued)

Low	High	Metric acronym	Definition of class-level metric
31	69	`min.patch.core.area`	Minimum patch core area
32	70	`max.patch.core.area`	Maximum patch core area
33	71	`prop.like.adjacencies`	Calculated from the adjacency matrix, which shows the frequency with which different pairs of patch types (including like adjacencies between the same patch type) appear side-by-side on the map (measures the degree of aggregation of patch types)
34	72	`aggregation.index`	Computed simply as an area-weighted mean class aggregation index, where each class is weighted by its proportional area in the landscape
35	73	`lanscape.division.index`	Based on the cumulative patch area distribution and is interpreted as the probability that two randomly chosen pixels in the landscape are not situated in the same patch
36	74	`splitting.index`	Based on the cumulative patch area distribution and is interpreted as the effective mesh number, or number of patches with a constant patch size when the landscape is subdivided into S patches, where S is the value of the splitting index
37	75	`effective.mesh.size`	Equals 1 divided by the total landscape area (m^2) multiplied by the sum of patch area (m^2) squared, summed across all patches in the landscape
38	76	`patch.cohesion.index`	Measures the physical connectedness of the corresponding patch type

Data Input Format

Currently, the Remmel–Fortin code is constrained to square binary landscapes having 64×64 pixels. While computation for larger landscapes is possible, processing times become prohibitive for demonstration purposes. Remember that some possible bias may result in computations for some LMs due to constraints on patch size using a 64×64 landscape. If importing your own landscapes for future analyses, ensure that all landscape representations are in numerical matrix format.

Step 0: Initialize Workspace by Loading Libraries, Demo Data, and Lookup Tables

If the package **`SDMTools`** has not been downloaded and installed on your computer, you will need to install it from the Comprehensive R Archive Network (CRAN) along with all dependencies. Then, load the library and source the code provided for this chapter.

```
> library(SDMTools)
> load("Remmel-Fortin.save")
```

EXERCISES

EXERCISE 1: Analysis of a Single Landscape

Using a single landscape, one can assess the significance of observed LMs relative to expectations based on null empirical distributions. The null hypothesis would be that the LM value could occur due to random chance, given the composition and configuration of that landscape. The expected distribution (and its variability) is fabricated by simulating landscapes with identical extent, spatial resolution, composition, and configuration, from which LM values would be quantified. If the observed LM falls within either tail of this distribution, it is then considered significantly different than what could be expected by chance.

In this exercise, you will compute a series of landscape metrics (see Table 7.1) on a landscape (represented as the `demoimage3` object). You will then determine whether observed LM values are significantly different from those based on null empirical distributions generated using the CAR simulator. We use $n = 100$ simulations to aid the feasibility of teaching as well as precedence (Remmel and Csillag 2003); though, this value could be adjusted in the code.

Use the following steps:

Step 1: Plot the Original Landscape Dataset

```
> plot.new()
> par(pty="s", mfrow=c(1,1))
> imaks(demoimage3)
> title("demoimage3")
```

Step 2: Compute Parameters from the Observed Binary Landscape

Two parameters need to be estimated from the input binary landscape data:

- The proportion of each category (black/LOW, white/HIGH)
- The estimated degree of spatial autocorrelation (recall the Whittle's algorithm)

Then, these two estimated parameters are used to generate the empirical distributions for 38 class-level metrics for both classes using $n = 100$ landscapes.

```
> result1 <- singlemap(IMG = demoimage3, VERBOSE = TRUE, reps = 100)
```

Notice the total number of landscape pixels is 4096 (64 × 64 pixels = 4096) or alternatively 2458 Black + 1638 White = 4096 total pixels. The proportion of black pixels is (2458/4096) = 0.60 and the corresponding proportion of white pixels is (1638/4096) = 0.40. The Whittle estimate of rho ($\rho_W = 0.00$) is zero, or very close to it, indicating the absence of spatial autocorrelation.

Step 3: Assess the Significance of Each Class-Level Metric

The significance of each metric can be assessed by computing its associated probability. In other words, the number of times the observed metric is greater or smaller than the empirical null distribution. The smallest probability that can be obtained is 0.01. Here, one can assess the significance of the LM by comparing the computed probability, *P*, against $\alpha = 0.05$. These results are returned with the `singlemap()` function call above.

Next, plot the observed metric values (as indicated by a red dot) for a select subset of the distributions of metrics (as specified by

`"metrics = c(2,7,18,20,21,22)"`; here 2 = `LOW.n.patches`, 7 = `LOW.edge.density`, 18 = `LOW.max.perim.area.ratio`, 20 = `LOW.sd.shape.index`, 21 = `LOW.min.shape.index`, 22 = `LOW.max.shape.index`).

NOTE: see Table 7.1 for how LMs link to these short-hand numbers. Recall that black, B, or LOW refers to the lower value of the two categories, and white, W, or HIGH for the other.

```
> singleplotter(data=result1, img = demoimage3, metrics=c
(2,7,18,20,21,22), rows=2, cols=3, addactual=TRUE, colour=TRUE)
```

If `addactual = FALSE`, then the red dot (indicating the original LM value being analyzed, shown within the simulated distribution) will not be added to each boxplot.

NOTE: Due to the stochastic nature of this approach, your results may differ from this, as well as differ from your classmates. Furthermore, `"NA"` in the results below indicates all simulated values were identical.

```
Actual Metric Value ( LOW.n.patches ):  3
88 higher values, 4 lower values, and 8 identical values as the map.
Probability of map having a value <= to expectation: P=0.9600
Probability of map having a value >= to expectation: P=0.1200

Actual Metric Value ( LOW.edge.density ):  0.9897461
25 higher values, 73 lower values, and 2 identical values as the map.
Probability of map having a value <= to expectation: P=0.2700
Probability of map having a value >= to expectation: P=0.7500

Actual Metric Value ( LOW.max.perim.area.ratio ):  4
0 higher values, 5 lower values, and 95 identical values as the map.
Probability of map having a value <= to expectation: P=0.9500
Probability of map having a value >= to expectation: P=1.0000

Actual Metric Value ( LOW.sd.shape.index ):  11.10245
5 higher values, 95 lower values, and 0 identical values as the map.
Probability of map having a value <= to expectation: P=0.0500
Probability of map having a value >= to expectation: P=0.9500
```

```
Actual Metric Value ( LOW.min.shape.index ):  1
1 higher values, 0 lower values, and 99 identical values as the map.
Probability of landscape having a value <= to expectation: P=1.0000
Probability of landscape having a value >= to expectation: P=0.9900

Actual Metric Value ( LOW.max.shape.index ):  20.23
22 higher values, 78 lower values, and 0 identical values as the map.
Probability of map having a value <= to expectation: P=0.2200
Probability of map having a value >= to expectation: P=0.7800
```

The probabilities above indicate how likely the observed map LM is relative to the simulated values (the null, empirical distribution). Two probabilities are given. First shown is the probability that the map value is less than or equal to the expected value obtained from the empirical null distribution. Second is the probability that the map value is greater than or equal to the expected value obtained from the empirical distribution. These values are computed as (e.g., `LOW.n.patches`): $P=(88+8)/100=0.9600$, where the 8 is the number of simulated values equal to the observed one, and 88 is the number of simulated values greater than the observed one. Thus, $P=0.9600$ that the observed LM value for the map is less than what would be expected from the empirical distribution (because there are many more simulated landscapes that had higher values).

These probabilities can be considered for one- or two-tailed tests, depending on the context of the question posed. If the question asks simply whether an observed LM computed for a map differs from an expected value, the test would be two-tailed, and it is possible to specify a probability level that identifies how far in the tail of the distribution the observed LM resides. However, if the question posed implies directionality (e.g., is this landscape more fragmented than expected?), the test would be one-tailed. Thus, the probability that of the LM is in the upper-tail would be assessed because the specified LM measuring fragmentation would have this as an implied directionality.

Step 4: Explore Additional Metrics

Try this exercise using different class-level metrics (e.g., select five new class-level LMs; see Table 7.1) and different landscape datasets (e.g., `demoimage1`, `demoimage2`, `demoimage3`, and `demoimage4`).

Q1 Are the class-level LMs significantly different than expected under the null hypothesis of random chance, given null empirical distributions constructed based on simulations with identical composition and spatial autocorrelation? You will need to run `singlemap()` and `singleploter()` functions for each landscape you wish to assess (Table 7.2).

Q2 Are the probabilities different than one would expect given how the original data were generated (see Figures 7.1 and 7.2)?

Q3 How do the probabilities differ from what one would expect based on the proportion of the classes in the landscapes (e.g., demoimage1 and demoimage2 or demoimage3 and demoimage4) versus the degree of spatial autocorrelation (e.g., demoimage1 and demoimage3 or demoimage2 and demoimage4) (see Figure 7.1)?

Table 7.2 Sample LM results for **demoimage3** with actual values and then probabilities in parentheses. The probabilities shown for **demoimage3** are probabilities of having an LM value less than or equal to expectations. Entries for **demoimage2** are to be filled by students

LM	demoimage3	demoimage2
Number of patches	3 (0.9600)	
Edge density	0.989746 (0.2700)	
Maximum perimeter/area ratio	4 (0.9500)	
Standard deviation shape index	11.10245 (1.0000)	
Minimum shape index	1 (1.0000)	
Maximum shape index	20.23 (0.2200)	

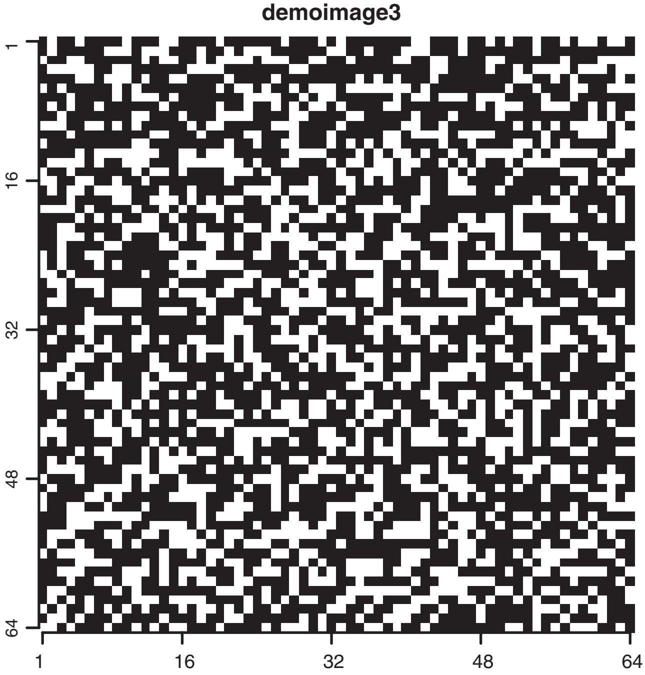

Figure 7.2 Plot of landscape **demoimage3** with two categories (LOW in black and HIGH in white) and 64×64 pixels

EXERCISE 2: Comparing Two Different Landscapes

For two different landscapes (either the same landscape at two different time periods, or two landscapes at different locations), it is also important to assess whether observed LM values are similar based on null empirical distributions. The null hypothesis is that the mean expected LM value is equal to the same LM from a simulated landscape with identical extent, spatial resolution, composition, and spatial autocorrelation. We will test the null hypothesis by assessing whether distributions for the specified LM overlap by a specified amount; if the overlap is large, the two values are not considered significantly different.

In this exercise, you will compute a series of landscape metrics (Table 7.1) on two landscapes (`demoimage2` and `demoimage3`) to determine whether the spatial structures of the landscapes are similar or not. You will learn how to determine whether or not class-level metrics from two landscapes are significantly different based on expectations from null empirical distributions ($n = 100$ simulated landscapes) generated using the CAR simulator. To do so, perform the following steps.

Step 1: Plot the Two Original Landscape Datasets (Figure 7.3)

```
> plot.new()
> par(pty="s", mfrow=c(1,2))
> imaks(demoimage2)
> title("demoimage2")
> imaks(demoimage3)
> title("demoimage3")
```

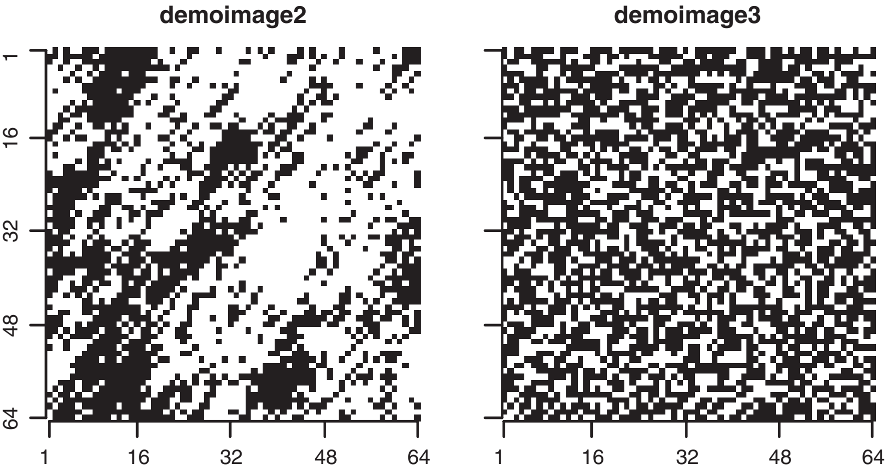

Figure 7.3 Plot of landscapes **`demoimage2`** and **`demoimage3`**

Step 2: Compute Parameters for Each Observed Binary Landscape

As in the first exercise, it is necessary to estimate pattern parameters for both input maps: (1) the proportion of each category (black, white), and (2) the estimated degree of spatial autocorrelation based on Whittle's algorithm (as in Appendix A). Then, the two estimated parameters are used to generate the empirical distributions for all the 38 class-level metrics for both categories using 100 simulated landscapes based on the CAR simulator (as described earlier).

```
> result1 <- singlemap(IMG = demoimage3, VERBOSE = TRUE,
reps = 100)
```

You should produce the same results here as in Step 2 of Exercise 1 (see explanation in Exercise 1).

```
> result2 <- singlemap(IMG = demoimage2, VERBOSE = TRUE,
reps = 100)
```

Step 3: Assess the Significance of Each Class-Level Metric

The significance of each class-level metric can be assessed by computing its probability (i.e., the number of times the observed class-level metric is greater or smaller than the expected metric under the empirical null distribution based on 100 replicates). Plot the observed class-level metrics (as indicated as a red dot) and a selected subset of the empirical distributions of the class-level metrics indicated as: "`metrics =c(2,7,18,20,21,22)`" where 2=`LOW.n.patches`, 7=`LOW.edge.density` and 18=`LOW.max.perim.area.ratio`, 20=`LOW.sd.shape.index`, 21=`LOW.min.shape.index`, 22=`LOW.max.shape.index`).

```
> singleplotter(data=result1, img=demoimage3, metrics=c(2,7,18,
20,21,22), rows=2, cols=3, addactual=TRUE, colour=TRUE)
> singleplotter(data=result2, img=demoimage2, metrics=c(2,7,18,
20,21,22), rows=2, cols=3, addactual=TRUE, colour=TRUE)
```

For explanation of these results, see Step 3 in Exercise 1.

Step 4: Plot Side-By-Side Boxplots

For comparison purposes, it is useful to produce side-by-side boxplots for each metric, contrasting the range of variability between map pairs. The range of LM variability comes from the *n* simulated landscapes, forming the empirical null distribution and metric expectation.

```
> doubleplotter(data1 = result1, data2 = result2, img1 = demoimage3,
img2 = demoimage2, metric = 8)
> doubleplotter(data1 = result1, data2 = result2, img1 = demoimage3,
img2 = demoimage2, metric = 15)
```

Step 5: Assess the Significance of Each Class-Level Metric

Assess whether there is a significant difference for each class-level metric for each observed landscape and interpret the results to determine whether the confidence intervals of each metric are overlapping. This can be most easily determined visually by looking at the boxplots for overlap of the notched region, but could be done numerically by extracting the simulated values stored in the result objects (where exact value ranges of the notches could be computed and compared) (Figures. 7.4 and 7.5).

Step 6: Additional Lab Exercises

Try this exercise using different class-level metrics (e.g., select five new class-level LMs; see Table 7.1) and different landscape datasets (e.g., demoimage1, demoimage2, demoimage3, and demoimage4).

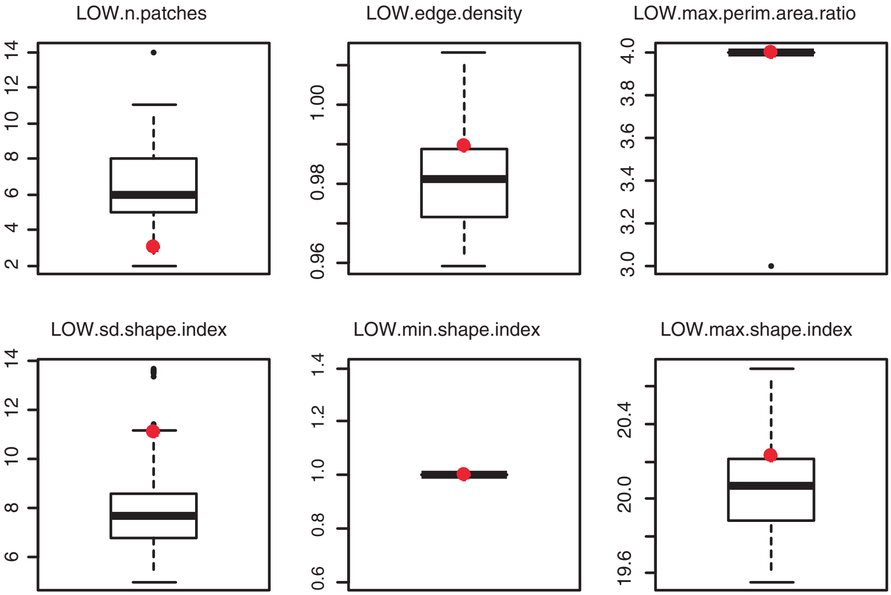

Figure 7.4 Boxplots of null empirical class-level metrics based on 100 simulated landscapes. The observed LM value is indicated by *red dot*

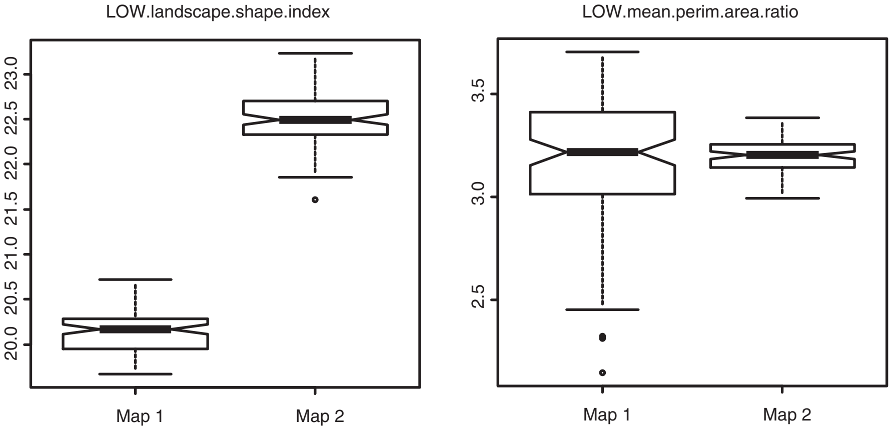

Figure 7.5 Boxplots of the class-level metrics for the two landscapes (Map 1 = **`demoimage3`** and Map 2 = **`demoimage2`**). *Left panel* shows confidence intervals for the landscape shape index and *right panel* shows mean perimeter area ratio. When the boxplot notches of two landscapes do not overlap, one can conclude that the two medians differ (Chambers et al. 1983)

Q4 Are the class-level LMs significantly different than expected under the null hypothesis of random chance, given the null empirical distributions constructed based on simulations with identical composition and spatial autocorrelation? You will need to run `singlemap()` and `singleplotter()` functions for each landscape you wish to assess. This is done individually for each landscape selected; when comparing two landscapes, it must be done individually for each of these two landscapes.

Q5 Are the probabilities different than one would have expected given how the original data were generated (see Figure 7.1)?

Q6 How do the probabilities differ from how one would expect according to the proportion of the classes in the landscapes (e.g., `demoimage1` vs. `demoimage2`, or `demoimage3` vs. `demoimage4`) compared with the degree of spatial autocorrelation (e.g., `demoimage1` vs. `demoimage3`, or `demoimage2` vs. `demoimage4`) (see Figure 7.1)?

Q7 For each pair of landscapes compared, and each LM you are interested about, are the landscapes significantly different based on the overlap of the boxplots (as was illustrated in Figure 7.5)?

EXERCISE 3: Determine a Landscape's Position Within the Distribution of Possible Class-Level Metric Values

Landscape pattern assessment can help inform an intervention or manipulation for management purposes. Restoring (or enhancing or adjusting) landscape spatial structure may be needed to influence provisioning of ecosystem services. In the context of conservation, one might wish to reduce fragmentation to help maintain biodiversity. To accomplish either, one must first determine the extent to which changing composition vs. spatial autocorrelation might change LM values. After deciding on one or more useful, robust, and informative LM(s), the next step would be to determine where the LMs reside within their null empirical distributions.

The goal of this third exercise is to determine where the observed landscape exists within the class-level metric space, given its proportion and estimated degree of spatial autocorrelation. Observing LM values among the joint influences of composition and configuration permits us to identify which aspect of spatial pattern (when manipulated) would most efficiently lead to the desired LM value change. To do so, perform the following steps.

Step 1: Perspective Plots of Class LM Median and Variance

Begin by producing a perspective plot for the median of a selected LM as it varies with class proportion and degree of spatial autocorrelation. The example provided has prop = 0.72 and $\rho_W = 0.49$ relative to a given class-level metric; the median value is based on 100 simulated landscape replicates.

```
> tempmed <- apply(surfaces[9,,,], MARGIN=c(1,2), median)
> persp(tempmed, ticktype="detailed", cex.axis=0.7, zlab="Metric",
ylab="Proportion", xlab="Rho", theta=-45)
```

The variance of this surface can also be computed to indicate the amount of variability at each point on the surface for a selected class LM along the identical axes of proportion of the category and the degree of spatial autocorrelation. The example below is for the ninth class metric: largest patch index for the black (or LOW) category

```
> tempvar <- apply(surfaces[9,,,], MARGIN=c(1,2), var)
> persp(tempvar, ticktype="detailed", cex.axis=0.7, zlab="Metric",
ylab="Proportion", xlab="Rho", theta=-45)
```

Step 2: Drop a Point Onto the Perspective Plot Indicating Observed Landscape's Position

Draw the perspective plot with a point indicating the location of a specific proportion and ρ relative to a single metric (Figure 7.6).

```
> surfplot(surfaceobj=tempmed, prop=0.72, rho=0.49, colour=TRUE,
drop=TRUE, cross=FALSE)
```

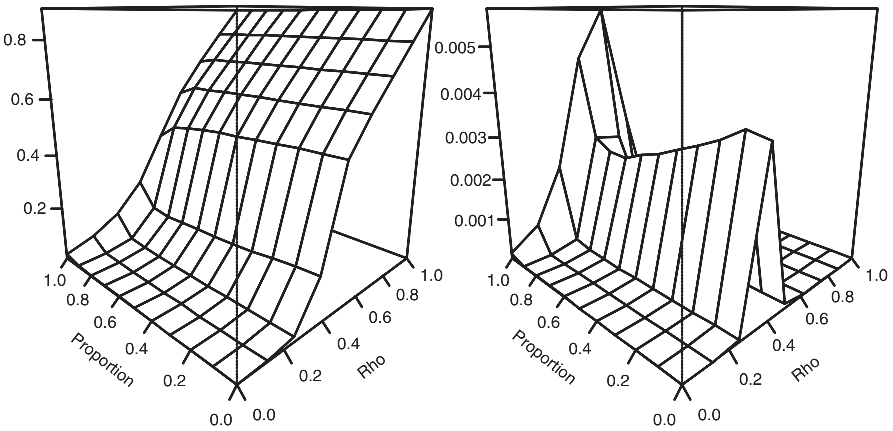

Figure 7.6 Surface response for class-level metric (z-axes; *left panel*: median, stored in `tempmed`; *right panel*: variance, stored in `tempvar`) according to proportion and spatial autocorrelation. These indicate how the expected value (median) and the variability (variance) of the largest patch index for the black (or LOW) category behave as composition and configuration parameters are incrementally changed

Step 3: Produce Boxplots of Surface Variability Along Axes Crossing Through the Observed Landscape Position Within the Perspective Plot

Include boxplots for Step 1 at the level of the observed proportion and spatial autocorrelation (ρ_W) axes on the surface to depict the variability (to do so set `cross=TRUE`).

```
> surfplot(surfaceobj=tempmed, prop=0.72, rho=0.49, colour=TRUE,
drop=TRUE, cross=TRUE)
```

Step 4: Perspective Plot with Both Median and Variance of Selected LM

Start by plotting the perspective plot of the median value of the class-level metric. This surface shows the variance (in color) indicating the variability at each point on the surface along axes of proportion and the degree of spatial autocorrelation (ρ_W).

The example below (Figure 7.7) demonstrates the ninth class metric: largest patch index for the LOW category. To color the median surface based on the variance values, the library *fields* must be installed and loaded along with dependencies.

```
> library(fields)
```

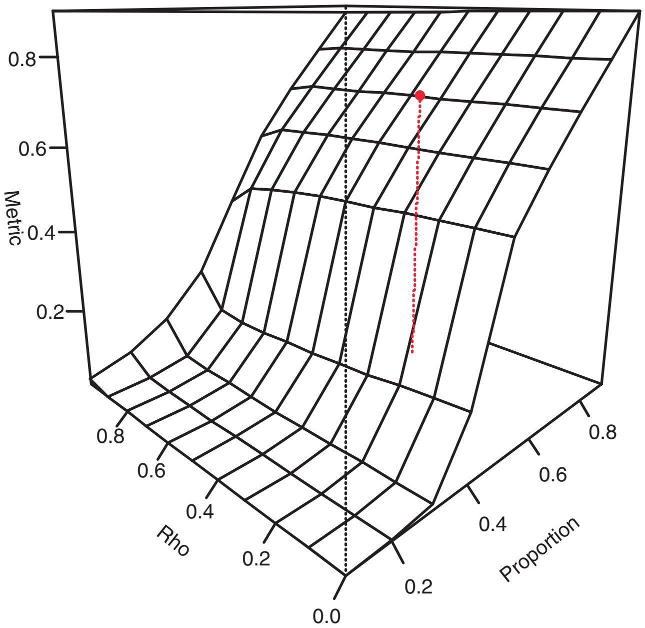

Figure 7.7 Surface response of the class-level metric median according to incremental variation of proportion and spatial autocorrelation for the category largest patch index as computed for the LOW value. The dropped red point indicates the position of the observed landscape within the range of variation that varying proportion and spatial autocorrelation can produce

Then, the plot can be produced:

```
> drape.plot(seq(0.1,0.9,by=0.1), seq(0,0.2499999, by=0.2499999/10)*4,
tempmed, tempvar, ticktype="detailed", col=topo.colors(50), theta=-25,
phi=15, cex.axis=0.5, xlab="Proportion", ylab="Rho", zlab="Metric")
```

Step 5: Additional Lab Exercises

Try this exercise using different class-level metrics (e.g., select five new class-level LMs; see Table 7.1) and different landscape datasets (e.g., `demoimage1`, `demoimage2`, `demoimage3`, and `demoimage4`).

Q8 Given the observed class-level LM value (i.e., the red dot in Figure 7.8) for the largest patch index within the context of all possible values of composition and spatial autocorrelation (Figures 7.8 and 7.9), what strategy (or strategies) would you recommend to reduce fragmentation (i.e., to increase the largest patch

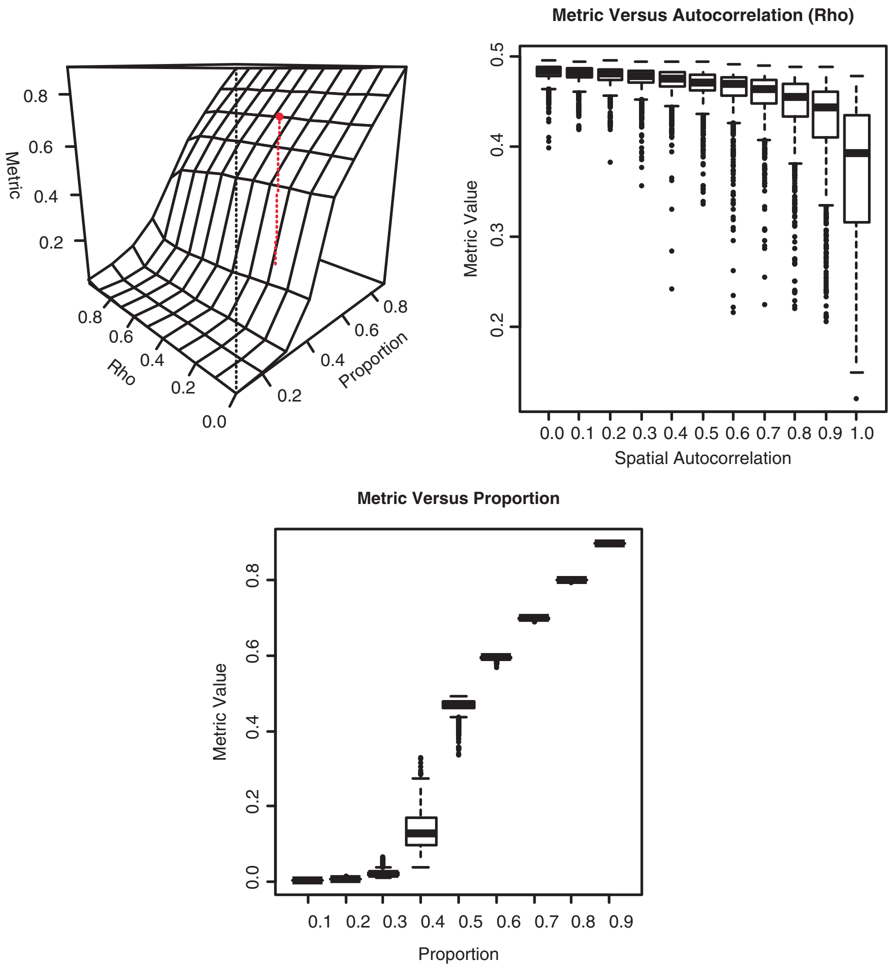

Figure 7.8 Class-level metric values according to category proportion and spatial autocorrelation: *left panel* as in Figure 7.6; middle panel: boxplot across spatial autocorrelation for the observed proportion level; *right panel*: boxplot across proportion level for the observed degree of spatial autocorrelation

index value)? Would it be easier to adjust the proportion or the level of spatial autocorrelation to achieve the goal? Imagine adding or removing land cover (either the white or black category) or simply rearranging them. Along which axis (composition or spatial autocorrelation) would LM values change more rapidly? How much change would be required to effect a 0.1 change to the LM value?

Q9 Repeat the previous question by examining another LM. You will need to begin at step 1 and adjust the value for the LM from 9 to the metric you select (Table 7.1).

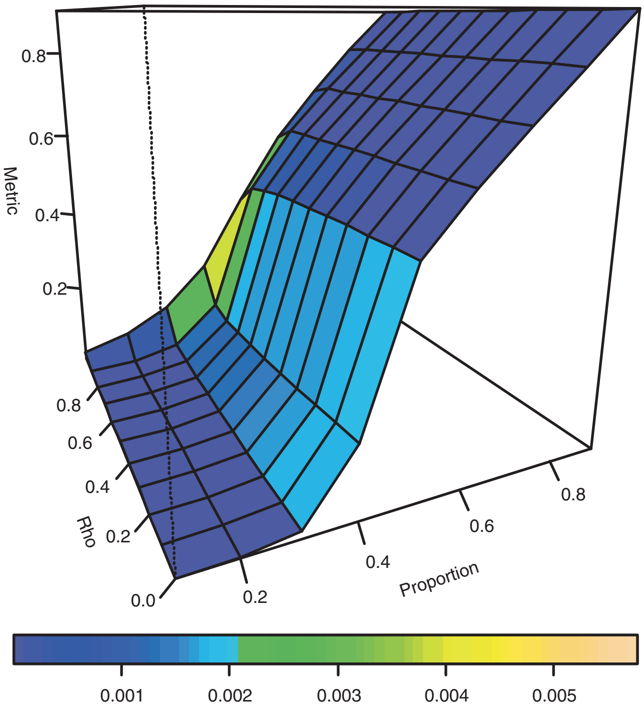

Figure 7.9 Class-level metric surface (as in Figure 7.8) combining the median LM value with the variability shown in colour across the ranges of proportion and configuration

Q10 Note and describe how the generated surfaces differ (along with the cross-sectional boxplots). Indicate the easiest way to decrease that LM value; would it be easier to alter composition or spatial autocorrelation?

CONCLUSIONS

While landscape metrics have been computed for decades, a major issue revolves around the difficulty in testing their significance. By using a simulation approach as presented here, it is now possible to assess whether or not the observed spatial structure of the landscape could have occurred by chance alone. Assessing the significance of LM is a major step in relating spatial structure to underlying process(es) that generated it. Furthermore, the knowledge of where in the range of proportion and spatial autocorrelation observed LM values lie can be used to propose restoration strategies for conservation purposes.

Acknowledgements We appreciate the contributions from Drs. Sándor Kabos and Ferko Csillag for their roles in providing some of the earlier programming work that we expanded upon. The chapter benefited from comments by R.H. Gardner, S.E. Gergel, M.G. Turner, and an anonymous reviewer.

APPENDIX A. EXPLANATION OF WHITTLE'S ALGORITHM (WHITTLE 1954)

For continuous data, ρ can be estimated using Whittle's algorithm (Whittle 1954) that extends the convention of time-series analysis to spatial processes reflected as collections of linear transects in geographic space. There is however the chance of bias in the estimated value of spatial autocorrelation when applying this algorithm to categorical data. This bias varies according to the composition, pi, such that around an even proportion of two classes, the bias is relatively small; however, it can be quite strong when the proportions differ greatly. Therefore, a correction factor needs to be applied to adjust the spatial autocorrelation estimate, resulting in the "true" ρ_{W} for categorical data. This true ρ_{W} needs to be multiplied by 4 to compensate for the isotropy of the algorithm and to scale the estimation to a range between 0 and 1. As this correction factor requires intensive computation, it has already been performed and stored as a lookup table in the provided Remmel–Fortin code as object DIFF50 and is used internally when estimating ρ_{W}.

REFERENCES AND RECOMMENDED READINGS[1]

Chambers JM, Cleveland WS, Kleiner B et al (1983) *Graphical methods for data analysis*. Wadsworth & Brooks/Cole, Pacific Grove

Cressie NAC (1993) *Statistics for spatial data*, revised edn. Wiley, New York

*Fortin MJ, Boots B, Csillag F et al (2003) On the role of spatial stochastic models in understanding landscape indices in ecology. Oikos 102:203–212. *This paper puts into perspective the use of stochastic models and simulations for studying complex ecological systems, specifically spatial patterns. The authors present a theoretical framework for the implementation of stochastic models to examine the confidence intervals around various landscape metrics.*

Fortin MJ, Dale MRT (2005) *Spatial analysis: a guide for ecologists*. Cambridge University Press, Cambridge

Fortin MJ, Jacquez GM, Shipley B (2012) Computer-intense methods. In: El-Shaarawi AH, Piegorsch WW (eds) *Encyclopedia of environmetrics*, 2nd edn. Wiley, Chichester, pp 489–493

Gardner RH, Urban DL (2007) Neutral models for testing landscape hypotheses. Landsc Ecol 22:15–29

*Haines-Young R, Chopping M (1996) Quantifying landscape structure: a review of landscape indices and their application to forested landscapes. Progr Phys Geogr 20(4):418–445. *One of few papers that provide a balanced summary of landscape pattern indices available for assessing spatial patterns. The authors elucidate uses and constraints on various metrics.*

[1] *NOTE:* An asterisk preceding the entry indicates that it is a suggested reading.

Hargrove WW, Hoffman FM, Schwartz PM (2002) A fractal landscape realize for generating synthetic maps. Conserv Ecol 6(1):2

Manly BFJ (2006) *Randomization, bootstrap and Monte Carlo methods in biology*. 3rd edn. CRC Press: Boca Raton, Florida

McGarigal K (2002) Landscape pattern metrics. In: El-Shaarawi AH, Piegorsch WW (eds) *Encyclopedia of environmetrics*, vol 2. Wiley, Chichester, pp 1135–1142

McGarigal K, Marks BJ (1995) FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station. Gen. Tech. Rep. PNW-GTR-351, p 122

Moran PAP (1950) Notes on continuous stochastic phenomena. Biometrika 37(1/2):17–23

Proulx R, Fahrig L (2010) Detecting human-driven deviations from trajectories in landscape composition and configuration. Landsc Ecol 25:1479–1487

R Development Core Team (2010) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R--project.org/

*Remmel TK, Csillag F (2003) When are two landscape pattern indices significantly different? J Geogr Syst 5(4):331–351. *The authors present an extensive simulation experiment that demonstrates how empirical distributions of landscape metrics can be used for statistically comparing spatial patterns expressed by binary gridded maps. The procedure improves on traditional hypothesis testes that assume normally distributed values for landscape metrics.*

*Remmel TK, Fortin M-J (2013) Categorical class map patterns: characterization and comparison. Landsc Ecol 28(8):1587–1599. *Provides much of the theoretical and statistical background and detail related to the exercises in this chapter. Here, the authors explore the use of landscape simulations to compare binary landscape patterns using class-level metrics.*

*Riitters KH, O'Neill RV, Hunsaker CT et al (1995) A factor analysis of landscape pattern and structure metrics. Landsc Ecol 10(1):23–39. *This keystone paper summarizes the redundancies present among many landscape pattern metrics and proposes new compound variables for characterizing landscape patterns. The new variables are produced by ordination of many commonly computed metrics.*

Turner MG, Gardner RH, O'Neill RV (2001) *Landscape ecology in theory and practice*. Springer, New York

Van Der Wal J, Falconi L, Januchowski S et al (2011) SDMTools: species distribution modelling tools: tools for processing data associated with species distribution modelling exercises. 1.1–5. The Comprehensive R Archive Network (CRAN). GPL 2.0, http://cran.r-project.org/

Whittle P (1954) On stationary processes in the plane. Biometrika 41:434–449

*Wu JG, Shen WJ, Sun WZ et al (2002) Empirical patterns of the effects of changing scale on landscape metrics. Landsc Ecol 17(8):761–782. *This pivotal paper examines the effect of spatial scale on landscape metrics. While scale is an important concept in ecology, the effects of grain size and extent are not always clearly understood in the analysis of spatial patterns.*