

## Should the Mantel test be used in spatial analysis?

Pierre Legendre<sup>1\*</sup>, Marie-Josée Fortin<sup>2</sup> and Daniel Borcard<sup>1</sup>

<sup>1</sup>Département de Sciences Biologiques, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, QC, Canada H3C 3J7; and <sup>2</sup>Department of Ecology & Evolutionary Biology, University of Toronto, Toronto, ON, Canada M5S 3G5

### Summary

1. The Mantel test is widely used in biology, including landscape ecology and genetics, to detect spatial structures in data or control for spatial correlation in the relationship between two data sets, for example community composition and environment. The study demonstrates that this is an incorrect use of that test.
2. The null hypothesis of the Mantel test differs from that of correlation analysis; the statistics computed in the two types of analyses differ. We examined the basic assumptions of the Mantel test in spatial analysis and showed that they are not verified in most studies. We showed the consequences, in terms of power, of the mismatch between these assumptions and the Mantel testing procedure.
3. The Mantel test  $H_0$  is the absence of relationship between values in two dissimilarity matrices, not the independence between two random variables or data tables. The Mantel  $R^2$  differs from the  $R^2$  of correlation, regression and canonical analysis; these two statistics cannot be reduced to one another. Using simulated data, we show that in spatial analysis, the assumptions of linearity and homoscedasticity of the Mantel test ( $H_1$ : small values of  $D_1$  correspond to small values of  $D_2$  and large values of  $D_1$  to large values of  $D_2$ ) do not hold in most cases, except when spatial correlation extends over the whole study area. Using extensive simulations of spatially correlated data involving different representations of geographic relationships, we show that the power of the Mantel test is always lower than that of distance-based Moran's eigenvector map (dbMEM) analysis and that the Mantel  $R^2$  is always smaller than in dbMEM analysis, and uninterpretable. These simulation results are novel contributions to the Mantel debate. We also show that regression on a geographic distance matrix does not remove the spatial structure from response data and does not produce spatially uncorrelated residuals.
4. Our main conclusion is that Mantel tests should be restricted to questions that, in the domain of application, only concern dissimilarity matrices, and are not derived from questions that can be formulated as the analysis of the vectors and matrices from which one can compute dissimilarity matrices.

**Key-words:** landscape ecology, landscape genetics, Mantel test, Moran's eigenvector maps (MEM), network analysis, numerical simulations, redundancy analysis, spatially structured data

### Introduction

The Mantel test was originally designed for analysing disease clustering in epidemiological studies. In that procedure, Mantel (1967) related a matrix of spatial distances and a matrix of temporal distances in a generalized regression approach. The procedure was expanded by Mantel & Valand (1970) to a non-parametric form of analysis of the relationship between two dissimilarity matrices computed from two sets of multivariate data concerning the same  $n$  individuals or sampling units. Since that paper, 'the procedure, known as the Mantel test in the biological and environmental sciences, includes any analysis relating two distance matrices or, more generally, two resemblance or proximity matrices' (Legendre 2000).

In biology, Sokal (1979) was the first to use Mantel tests to study patterns of geographic variation in taxonomic data. In Sokal & Rohlf's (1995) *Biometry* book, the Mantel test is presented as a general procedure to test the relationship between multivariate data tables expressed as dissimilarity matrices in

biological problems; for these authors, the usefulness of the Mantel test derived from the fact that 'in evolutionary biology and ecology, dissimilarity coefficients are frequently used to measure the degree of difference between individuals, populations, species, or communities' (Sokal & Rohlf 1995, p. 813). A further generalization was proposed by Anselin (1995) who showed that indices of spatial autocorrelation such as Moran's  $I$  and Geary's  $c$  may be considered to be special cases of the Mantel statistic.

The discussion and criticisms formulated in this study only concern the spatial analysis applications of the Mantel test in biology (ecology, genetics, evolutionary biology, landscape ecology and landscape genetics). They do not concern the original test developed by Mantel for epidemiological studies, where the question clearly involved the relationship between two types of distances (temporal and spatial) separating disease occurrences.

Applications to spatial analysis started when ecologists and geneticists discovered that a Mantel test offered an easy way of introducing spatial relationships, in the form of a geographic distance matrix, into a statistical framework for modelling

\*Correspondence author. E-mail: Pierre.Legendre@umontreal.ca

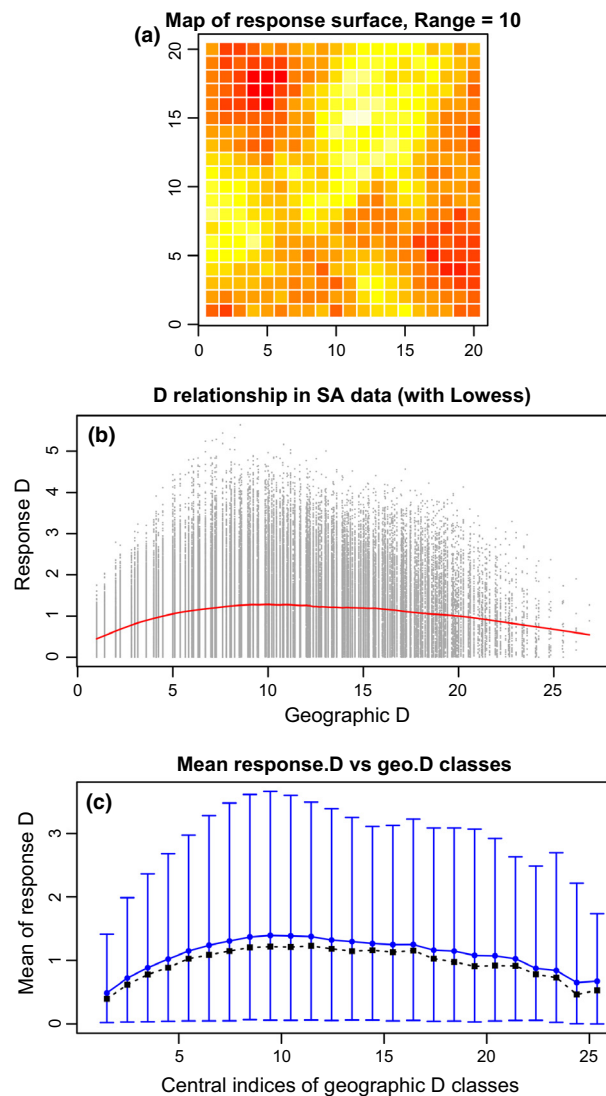
multivariate data (Sokal 1979; Legendre & Troussellier 1988; Legendre & Fortin 1989; Cushman *et al.* 2006). The Mantel test quickly became a favourite statistical procedure for researchers interested in spatial [or temporal] processes. That was before more appropriate and powerful statistical procedures, such as dbMEM analysis, used in the simulations reported in this study, became available; see Legendre, Borcard & Peres-Neto (2005), Legendre & Fortin (2010) and Dray *et al.* (2012).

The main thesis of this study is that Mantel tests should be restricted to questions that, in the domain of application, only concern dissimilarity matrices, and are not derived from questions that can be formulated as the analysis of 'raw data tables', meaning the vectors and matrices from which one can compute dissimilarity or distance matrices. Matrices of geographic distances among sites derived from spatial coordinates are included in the cases where Mantel tests may be inappropriate. (i) We will show that the hypotheses of correlation tests of significance of raw data tables differ from the hypotheses that concern dissimilarity matrices; furthermore, the statistics involved in the two types of analyses differ and cannot be reduced to one another. (ii) We will refer to simulation papers that have shown that analyses in the world of *raw data* are consistently more powerful than in the world of *dissimilarities* when both approaches are possible. Appendix S1 (Supporting Information) retraces the history of the applications of the Mantel test to spatial data analysis and summarizes the most important simulation studies that have shown that the approach lacks statistical power by a broad margin. (iii) We will focus on the basic assumptions of linearity and homoscedasticity of the Mantel test in spatial analysis. Simulations involving spatially autocorrelated data will show that these assumptions are not verified in most studies. (iv) Finally, using again simulations of spatially autocorrelated data, we will show the consequences, in terms of power, of the mismatch between these assumptions and the Mantel procedure.

Formally, a *dissimilarity index* (or *coefficient*) is a function that measures the difference between two vectors. A *distance index* is a special type of dissimilarity that satisfies the metric properties (minimum value of 0, positiveness, symmetry and triangle's inequality); the Euclidean distance is the most widely used distance coefficient. In this study, the general term *dissimilarity* will be used except to designate a spatial or temporal distance.

### What is the null hypothesis of the Mantel test?

Scientists who use Mantel tests when the analysis of raw data tables is possible are usually under the impression that the two types of methods are testing the same statistical hypothesis. For example, Guillot & Rousset (2013) wrote in the caption of their Fig. 2: 'The null hypothesis tested [in the Mantel test] is the independence between  $x$  and  $y$ ' (in that part of their paper,  $x$  and  $y$  are two random variables, cf. their Fig. 1; they are not dissimilarity matrices). That description of the null hypothesis would be correct for the test of a correlation coefficient



**Fig. 1.** (a) Map of a  $20 \times 20$  pixel simulated autocorrelated surface. The variogram range controlling the autocorrelation structure was 10. Colour scale: from dark red (low) to pale yellow (high values). (b) Relationship between geographic distances ( $D$ ) among pixels (abscissa) and dissimilarities (unsigned differences) computed from the simulated data (response  $D$ ) whose values are represented by colours in panel a. This graph contains  $(20^2(20^2 - 1)/2) = 79\,800$  points (pairs of dissimilarities). Because of point superposition, the  $D$ - $D$  relationship central tendency is not clear; a Lowess smoother (red line) was added to indicate the central tendency of the relationship across the plot. (c) The geographic distances in the abscissa of panel b are replaced by distance classes; the central tendencies of individual values of the response dissimilarities within classes are represented by their means (blue circles) and medians (black squares); variation is represented by empirical 95% coverage intervals, that is, intervals containing 95% of the response dissimilarities in the class.

between two random variables. It is incorrect, however, for the Mantel test, which is a test of the absence of relationship *between the dissimilarities* in two dissimilarity matrices. A correct formulation of  $H_0$  for the Mantel test is the following: ' $H_0$ : The distances among objects in matrix  $D_Y$  are not (linearly or monotonically) related to the corresponding distances in  $D_X$ ' (Legendre & Legendre 2012, p. 600; italics added for emphasis).

sis). Similar formulations of the Mantel null hypothesis are found in Legendre (2000, p. 41): ‘The simple Mantel test is a procedure to test the hypothesis that *the distances* among objects in a [distance] matrix **A** are linearly independent of *the distances* among the same objects in another [distance] matrix **B**’ and in Legendre & Fortin (2010, p. 835). In partial Mantel test, mentioned in section ‘Assumptions of the Mantel test’,  $H_0$  states that  $\rho(\mathbf{AB.C}) = 0$ , where **A**, **B** and **C** are dissimilarity matrices (Legendre 2000).

A complementary point is the demonstration by Legendre & Fortin (2010) (their eqs. 1, 2 and 9) that the statistic used in the Mantel test is unrelated to that used to test the  $R^2$  statistic in [multiple] linear regression or redundancy analysis (RDA), or the simple correlation coefficient  $r$ . Here, we highlight the difference between the  $R^2$  statistics tested in redundancy analysis (which is the multivariate form of multiple linear regression) and in a Mantel test.

In [multiple] linear regression and RDA,  $R^2$  is the ratio of the sum of squared differences from the mean, or sum of squares (SS) for short, of the fitted values to the sum of squares of the data:

$$R^2 = \frac{SS(\hat{\mathbf{Y}})}{SS(\mathbf{Y})} \quad \text{eqn 1}$$

(following usual notation), whose denominator is

$$SS(\mathbf{Y}) = \sum_{j=1:p} \sum_{i=1:n} (y_{ij} - \bar{y}_j)^2 \quad \text{eqn 2}$$

where  $n$  is the number of observations and  $p$  is the number of variables in matrix **Y**. This denominator can also be written as

$$SS(\mathbf{Y}) = \left( \sum_{i>h} D_{ih}^2 \right) / n \quad \text{eqn 3}$$

Proof of this equivalence is found in Appendix A1 of Legendre & Fortin (2010).  $R^2$  represents the fraction of the total sum of squares of the response data **Y** that is explained by the explanatory variables **X**.

Consider now two dissimilarity matrices, **D<sub>Y</sub>** and **D<sub>X</sub>**, computed from data vectors **y** and **x** or from matrices **Y** and **X**. String out the lower diagonal portions of these matrices as long vectors **d<sub>Y</sub>** and **d<sub>X</sub>**, each of length  $n(n-1)/2$ . The Mantel correlation,  $r_M$ , is the correlation coefficient between these two vectors. The square of  $r_M$  is the coefficient of determination  $R_M^2$  of the linear regression of **d<sub>Y</sub>** on **d<sub>X</sub>**:

$$R_M^2 = \frac{SS(\hat{\mathbf{d}_Y})}{SS(\mathbf{d}_Y)} \quad \text{eqn 4}$$

The denominator of that equation is

$$SS(\mathbf{d}_Y) = \sum_{i>h} (D_{ihY} - \bar{D}_Y)^2 = \sum_{i>h} D_{ihY}^2 - \frac{(\sum_{i>h} D_{ihY})^2}{n(n-1)/2} \quad \text{eqn 5}$$

This formula is written using dissimilarity values  $D_{ih}$  to make it comparable to Eqn 3. The important point here is that  $SS(\mathbf{d}_Y)$  in Eqn 5 is not equal to, is not a simple function of, and cannot be reduced to  $SS(\mathbf{Y})$  in Eqn 3. They are different statistics, and so are  $R^2$  and  $R_M^2$ .

The statistic used in each test reflects its null hypothesis and, because the null hypotheses differ, the statistics also differ and are not interchangeable. Hence, these two tests are not equivalent. This demonstration completes our proof that the Mantel test is inappropriate to test a hypothesis of correlation between two data vectors or matrices of raw data.

## Assumptions of the Mantel test

The Mantel test makes two strong assumptions about the relationships between the two sets of dissimilarities, **D<sub>1</sub>** and **D<sub>2</sub>**, under comparison.

The first assumption is that the relationship is linear, if a cross-product or a linear correlation coefficient is used as the Mantel statistic, or monotonic if the dissimilarities are replaced by their ranks (Mantel 1967) or if a Spearman or Kendall correlation coefficient is used to compute the Mantel statistic (Dietz 1983). The linearity or monotonicity assumption is linked to the choice of the statistic.

The second assumption, which is the basis for the alternative hypothesis ( $H_1$ ) of the Mantel test, is that small values of **D<sub>1</sub>** correspond to small values of **D<sub>2</sub>** and large values of **D<sub>1</sub>** to large values of **D<sub>2</sub>**. Mantel stated this assumption (alternative hypothesis  $H_1$ ) as follows in his 1967 paper (p. 209) in the context of the disease clustering problem: ‘if there is time-space clustering, cases in a cluster will be close both in time and space, while unrelated cases will tend to have a larger average separation in time and space’. In their *Biometry* textbook, Sokal & Rohlf (1995, pp. 814 and 816) formulated in similar terms the alternative hypothesis of the Mantel test for specific biological examples.

This assumption may hold for space-time clustering of epidemiological data, but does it hold for the various extensions of the Mantel test that are currently done by biologists? We will show in section ‘Simulations involving spatially autocorrelated data: violation of the Mantel test assumptions’ that for spatial analysis involving spatially autocorrelated data, that assumption, which refers to the homoscedasticity of the distribution of values in the distance–distance (**D–D**) plot, holds in a very limited number of situations; what is found in most cases is a hump-shaped or triangular distribution. This is a novel contribution to the Mantel debate.

## Misuse of the Mantel test to analyse georeferenced data

In many applications, researchers incorrectly used the Mantel and partial Mantel tests to assess hypotheses of relationships between variables or data tables, *not* between dissimilarity matrices. A list of examples is found in Legendre, Borcard & Peres-Neto (2005, pp. 438–439). Based on the demonstration reproduced in section ‘What is the null hypothesis of the Mantel test?’ and on numerical simulations, Legendre & Fortin (2010) argued that Mantel and partial Mantel tests should only be used to test hypotheses that specifically concern dissimilarities, not those derived artificially from hypotheses about the raw data. In particular, to test the correlation between two spa-

tially correlated vectors or matrices of raw data, one cannot use a partial Mantel test computed after transforming the raw data into dissimilarity matrices **A** and **B** and test  $H_0: \rho(\mathbf{AB.C}) = 0$ , where **C** is some form of geographic distance or connexion matrix. There are alternative ways of testing the significance of the correlation between two raw data vectors or matrices while controlling for spatial structure, as shown in Peres-Neto & Legendre (2010) and in Legendre & Legendre (2012).

All simulation studies carried out to measure the capacity of the partial Mantel test to control for (auto)correlation in data have been done by generating raw data that were spatially correlated, for example Manly (1986), Oden & Sokal (1992), Legendre, Borcard & Peres-Neto (2005), Legendre & Fortin (2010), Guillot & Rousset (2013) and section ‘Simulations involving spatially autocorrelated data: comparison of Mantel test and dbMEM analysis’ of this study. Throughout, the Mantel test was consistently shown to have low power in these simulations, compared to analyses performed on the original data. Appendix S1 reviews some of the papers that showed, through simulations, important characteristics of tests of significance in the presence of spatial correlation, including Mantel and partial Mantel tests.

### Simulations involving spatially autocorrelated data: violation of the Mantel test assumptions

Spatially autocorrelated surfaces of different sizes and degrees of autocorrelation were generated by Gaussian random field simulations, using function `RFsimulate()` of package **RandomFields** (Schlather *et al.* 2014) in R, implementing a spherical variogram model through function `RMspheric()`. Preliminary results, generated on a small surface ( $20 \times 20$  pixels), will be examined first.

Then, larger surfaces were generated in the same way and a subset of points was sampled: on each surface, we selected 100 points forming a square regular grid surrounded by 5-pixel-wide unsampled bands to reduce border effects in the sampled data. The points of the grid were spaced by 1–5 pixels; counting the border bands, the surfaces had {20, 29, 38, 47, 56} pixels in the horizontal and vertical directions, depending on the horizontal and vertical spacing {1, 2, 3, 4, 5} of the sampled points. Results for 5-pixel spacing will be examined. Similar (unreported) results were obtained for the smaller surfaces with horizontal and vertical spacing of 1–4 pixels. The results indicate the following about the assumptions of the Mantel test:

**1. Linearity assumption of the **D-D** comparison** – Let us examine first the response surface simulated on the ( $20 \times 20$  pixels) grid with spacing = 1 pixel and autocorrelation range = 10 units (Fig. 1a). The Lowess line in Fig. 1b and the response to distance classes in Fig. 1c show that the dissimilarities increased from geographic distance class 1 to 9 in this example; this is close to the range value (10) of the controlling variogram. The mean of the response dissimilarities decreased as geographic distance increased further. Hence, the **D-D** relationship was not linear or monotonic. Similar results are shown in Appendix S2 for larger ( $56 \times 56$  pixels) surfaces generated with different variogram range values. The only case

where the **D-D** relationship was approximately linear was that with range = 70 (Fig. S1q–r), where the autocorrelation range was near the maximum distance between pixels on the surface (i.e. between the pixels in opposite corners, whose geographic distance was 79.2 units).

**2. Assumption that small values of **D**<sub>1</sub> correspond to small values of **D**<sub>2</sub>, and large values of **D**<sub>1</sub> to large values of **D**<sub>2</sub>** – We will examine whether this assumption holds at least within the sections of the **D-D** plots within the range of the controlling variogram. This is the portion between geographic distances 1 and 9 or 10 in Fig. 1b,c. The graph shows that whereas small values of **D**<sub>1</sub> (response) correspond to small values of **D**<sub>2</sub> (geographic), an increasingly broad range of response values is associated with larger geographic distances, causing heteroscedasticity in the **D-D** distribution. The same absence of homoscedastic **D-D** relationships is found for the larger surfaces simulated with various range values (Fig. S2.1). The **D-D** relationship on the left of the geographic distance marking the end of the range of autocorrelation of the simulated surface is hump-shaped or triangular and, in any case, very far from homoscedasticity.

For spatially autocorrelated data, these two assumptions of the Mantel test are violated and that partly explains its lack of power. The violations are less important when autocorrelation is equal to or larger than the size of the study area; that is the case where the Mantel test performs best in terms of power, as we will see in the next section.

These two assumptions do not apply to the Mantel correlogram (Oden & Sokal 1986; Sokal 1986; Borcard & Legendre 2012) where the response dissimilarities **D**<sub>1</sub> are analysed in separate tests against a set of binary model matrices, each representing a geographic distance class.

### Simulations involving spatially autocorrelated data: comparison of Mantel test and dbMEM analysis

Despite several papers based on numerical simulations advising to the contrary (Appendix S1), the Mantel test is still widely used by ecologists and geneticists to carry out different forms of spatial analyses. That incentive led us to compare the power of the Mantel test to that of a test based on the original (non-dissimilarity) data, using extensive simulations carried out on the largest spatially autocorrelated surfaces of the previous section.

Spatially autocorrelated data were generated using function `RFsimulate()`, as in section ‘Simulations involving spatially autocorrelated data: violation of the Mantel test assumptions’. The following statistical methods will be compared to study the relationship between the values associated to the points and their geographic positions: (i) the Mantel test between dissimilarity matrices (with one-tailed tests in the upper tail; `mantel()` function of the **vegan** package, Oksanen *et al.* 2013) and (ii) spatial eigenfunction analysis using the form known as distance-based Moran’s eigenvector maps (dbMEM) (`PCNM()` function of the **PCNM** package, Legendre *et al.* 2012). That method is detailed in Legendre & Legendre (2012, Chapter 14) and in the original publications (Borcard & Legendre 2002;



Borcard *et al.* 2004; Dray, Legendre & Peres-Neto 2006) where it was called PCNM analysis. Spatial eigenfunctions can be used in linear models in the same way as any other set of explanatory variables. The analysis involves multiple linear regression when the response data are univariate (as in our simulation study) or redundancy analysis (RDA, Rao 1964) when it is multivariate. In both cases,  $R^2$  and adjusted  $R^2$  statistics ( $R^2_{adj}$ ) can be computed and tested for significance using a parametric or permutational  $F$ -test (Legendre, Oksanen & ter Braak 2011). A permutational test based upon 999 random permutations of the response data will be used. No variable selection will be carried out in this study; the analyses will be based upon the whole set of eigenfunctions that model positive spatial correlation, that is, those with positive Moran's  $I$  coefficients.

In all simulations, 1000 random autocorrelated surfaces with  $56 \times 56$  pixels were independently produced with variogram ranges of {0, 5, 10, 15, 20, 25, 30, 35, 40} grid units. These surfaces were sampled at 100 points forming a square regular grid with horizontal and vertical spacing of 5 units.

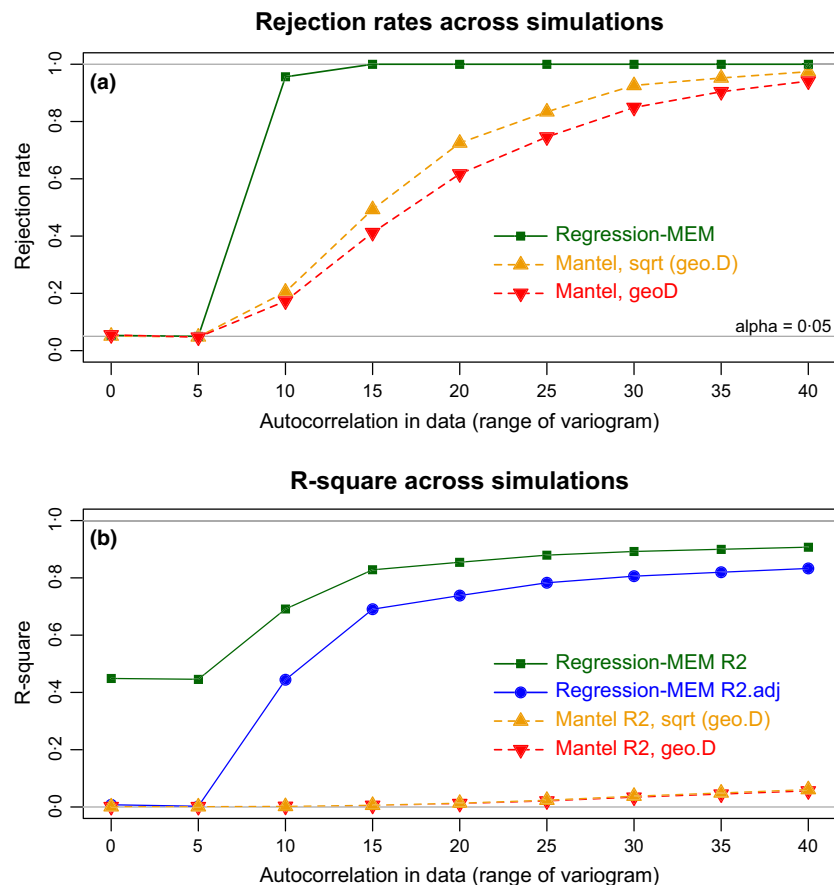
#### SERIES 1 SIMULATIONS INVOLVING ALL PAIRWISE GEOGRAPHIC DISTANCES

The simulated data sets were analysed with respect to geography using a dbMEM regression and a Mantel test. The truncation value for dbMEM generation was the point spacing, 5 grid units. Users of the Mantel test often square-root the geographic distances to increase the linearity of the relationships with the

response dissimilarities, so we carried out our study using both the original and square-rooted geographic distances.

For each range value, the 1000 simulation results were summarized by tallying how many data sets produced significant dbMEM and Mantel results at the  $\alpha = 0.05$  significance level (one-tailed tests in the upper tail); these numbers were divided by 1000 to obtain rejection rates, which were plotted against the variogram range values (Fig. 2a). Confidence intervals, based on the binomial distribution, were also computed. They are not visible in the graph because they were smaller than the symbols representing the rejection rates.

Each dbMEM regression produced an  $R^2$  and an  $R^2_{adj}$  statistic. The means of these  $R^2$  and  $R^2_{adj}$  across 1000 simulations were computed for each variogram range value. The means were actually computed on  $R^2$  transformed to  $\sqrt{1 - R^2}$  and  $\sqrt{1 - R^2_{adj}}$ , which have symmetric distributions, and transformed back to  $R^2$  and  $R^2_{adj}$ . Each Mantel test produced an  $r_M$  statistic, which was transformed to  $R^2_M$  by squaring it; with this transformation, the Mantel test is considered to be a form of regression analysis, following Mantel (1967). Many users of the Mantel test use that  $R^2_M$  statistic and erroneously interpret it as if it were equivalent to an  $R^2$  computed by regression on the raw data. Note that there is no way of adjusting  $R^2_M$  to account for the number of explanatory variables in matrix  $D_2$ . Means of the  $R^2_M$  values were computed as for the dbMEM  $R^2$  and  $R^2_{adj}$ . The mean  $R^2$  statistics were plotted against variogram range values (Fig. 2b) together with the mean  $R^2_{adj}$  statistics of dbMEM regression.



**Fig. 2.** (a) Rejection rates (i.e. number of rejections of  $H_0$  at the 0.05 significance level divided by the number of simulations, 1000) of the regression–dbMEM and Mantel tests as a function of the variogram range in the simulated data. (b) Mean  $R$ -squares of the two methods of analysis. The mean adjusted  $R$ -square ( $R^2_{adj}$ ) of the regression–MEM test, which is an unbiased estimate of the explained variation, is also shown. No  $R^2_{adj}$  statistic is available for Mantel tests.

The results (Fig. 2a) show first that the dbMEM analysis and Mantel test had correct levels of type I error; type I error was the rejection rate when there was no spatial autocorrelation in the data (range = 0) or when the range of the variogram used for generation of the data was not larger (range = 5) than the interval between the sampled grid points (here 5 units). This first result has been reported in other papers, for example Oden & Sokal (1992), Legendre & Fortin (2010) and Guillot & Rousset (2013).

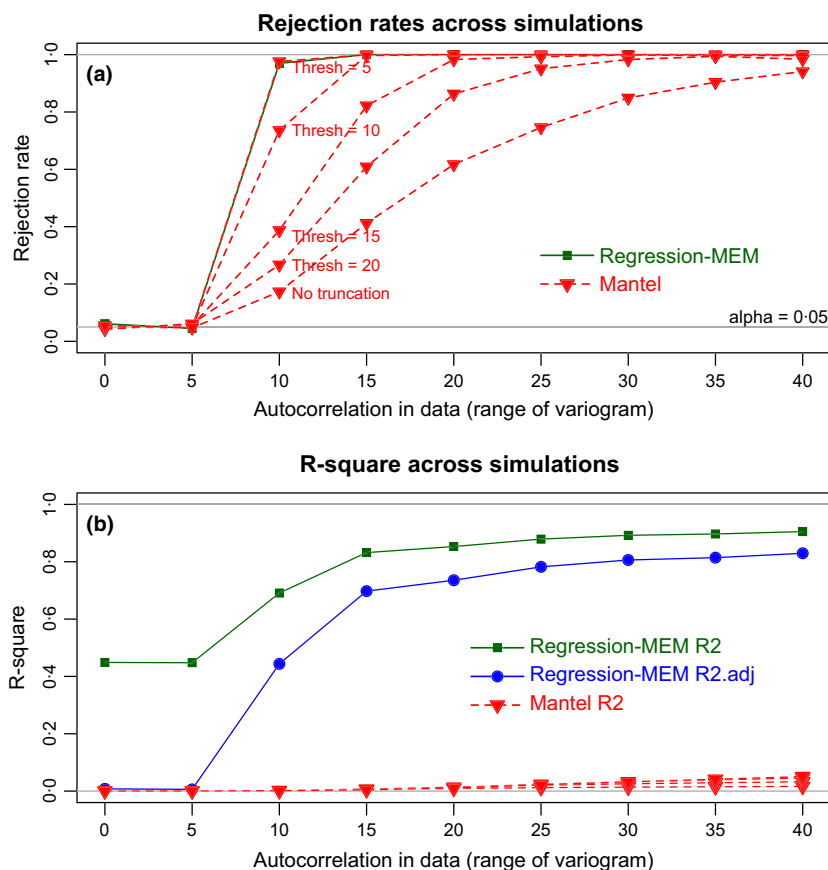
When the range of the variogram controlling the autocorrelation in the data was larger than 5, dbMEM analysis was always far more powerful than the Mantel test (Fig. 2a). When the range of the autocorrelation process became very large and the patches nearly covered the whole surface (Fig. S2.1k), the Mantel test became usable although its power remained lower than that of dbMEM analysis. In all cases, the Mantel test based on square-rooted geographic distances was slightly more powerful than the Mantel test based on untransformed geographic distances.

In a regression context,  $R^2$  is a useful measure of the variation of a response variable explained by explanatory data. Fig. 2b shows that the Mantel test  $R^2$  ( $R_M^2$ ) was much smaller than that of dbMEM regression. These two statistics are not comparable: in dbMEM analysis,  $R^2$  measures how much of the variance of the response data is explained by geography. In the Mantel test, it measures the fraction of the variance of the dissimilarities  $\mathbf{D}_1$  explained by the geographic distances  $\mathbf{D}_2$ . Hence, the Mantel  $R_M^2$  cannot be interpreted as

an estimate of the  $R^2$  produced by an analysis of the original data.

In our simulation functions, Mantel tests produced one-tailed tests in the upper tail. This is the normal output of **vegan**'s `mantel()` function and it was adequate for our study, where we wanted ( $H_1$ ) to detect positive spatial autocorrelation (SA) in the simulated data when SA was present. We checked, however, what happened in the lower tail. In simulations with variogram ranges of {0, 5}, there was no SA in the data because the spacing between points on the sampled grid was 5; as expected, the rejection rates in the upper and lower tails were always near the significance level, 0.05. When there was SA in the simulated data, the rejection rate in the upper tail increased, as shown in Figs 2–3, while it decreased and became 0 in the lower tail (not shown in the figures). It never went above the significance level.

Legendre & Fortin (2010, their Fig. 4) showed complementary results. They simulated a univariate regular gradient crossing a square map diagonally and added error (noise) to the response data. As the amount of noise increased, power of the methods of analysis decreased, as expected. The comparison involved a linear regression of the response data on the geographic coordinates of the sampled points (i.e. a linear trend surface analysis) and a simple Mantel test. The Mantel test became non-significant after a small amount of noise was added, whereas the  $F$ -test of the linear regression remained significant for higher amounts of noise. So in that example again, linear regression had higher power than the Mantel test.



**Fig. 3.** (a) Rejection rates (i.e. number of rejections of  $H_0$  at the 0.05 significance level divided by the number of simulations, 1000) of the regression–dbMEM and Mantel tests as a function of the variogram range in the simulated data. Mantel tests were computed with truncation levels (*thresh* in figure) of 5, 10, 15 and 20 grid units. (b) Mean  $R$ -squares of the two methods of analysis. The mean adjusted  $R$ -square ( $R_{adj}^2$ ) of the regression–MEM test, which is an unbiased estimate of the explained variation, is also shown. No  $R_{adj}^2$  statistic is available for Mantel tests. For the Mantel test, the  $R$ -squares obtained in the truncated and untruncated simulations are nearly identical and superposed in the figure.

## SERIES 2 SIMULATIONS INVOLVING TRUNCATED GEOGRAPHIC DISTANCE MATRICES

In these simulations, the matrix of geographic distances used in Mantel tests was truncated at different levels (thresholds, abbreviated *thresh* = {5, 10, 15, 20} grid units) and all distances larger than the truncation value were changed to the largest distance in the data set, which was the distance between the two opposite corners of the square grid (63.64 units). For each simulation condition (range and *thresh*), the analysis was repeated for 1000 independently generated surfaces. These simulations reproduced the method used by landscape ecologists and geneticists who apply Mantel tests to truncated distance matrices when they feel that the effect of the distance among sites can only be perceived up to a certain distance where contagion, dispersal of propagules in plants, or migration in animals, no longer creates spatial correlation among the sites (Dyer & Nason 2004; Fortuna *et al.* 2009; Murphy *et al.* 2010).

The truncated data, each with 100 observations, were analysed with respect to geography through a dbMEM regression using the full set of eigenfunctions modelling positive spatial correlation, as in series 1, and a Mantel test using the truncated geographic distance matrix (previous paragraph).

Rejection rates of the tests across the simulations are presented in Fig. 3a. For variogram ranges of 0 and 5, where there was no autocorrelation in the data, all tests had correct type I error as their rejection rates were close to the significance level. When the range was larger than 5, dbMEM analysis was always more powerful than the Mantel test for different truncation distance values (*thresh* in the figure), except when the truncation value was 5. The extreme case, with no truncation of geographic distances (or *thresh* larger than the largest distance in the data set), corresponds to the results in Fig. 2a. Hence, when more of the distances are kept (i.e. not truncated) in the geographic matrix, the Mantel test has less power to detect SA in the response data.

That the Mantel test with *thresh* = 5 had power identical to dbMEM analysis may seem surprising. This is because the geographic matrix only contained two different values in that case:  $D = 5$  for points that were at that distance, and the largest distance in the data set,  $D = 63.63961$ , for all other pairs of points. This was equivalent to the binary distance matrix used to test for autocorrelation in the first distance class of a Mantel correlogram. Our results thus show that the Mantel test used in this manner, with a single distance class, has the same power for detection of spatial autocorrelation as the dbMEM method of analysis. The simulation study of Borcard & Legendre (2012) had already shown that the test of significance in multivariate Mantel correlograms had high power. That is fine but it does not qualify the Mantel test as the equivalent of dbMEM analysis, which was developed to model the geographic distribution of univariate or multivariate data at different spatial scales, in addition to the production of a test for the presence of spatial correlation in data. In any case, when researchers use Mantel tests with truncated distance matrices, they have a specific ecological or genetic dispersion model in mind and they

do not truncate to keep only the first distance class. More about this in the Discussion.

The  $R^2$  results (Fig. 3b) tell the same story as reported in Fig. 2b: the square of the Mantel correlation ( $R_M^2$ ) is always extremely low.

## SERIES 3 SIMULATIONS INVOLVING DELAUNAY TRIANGULATIONS

In the interest of space, simulations involving Delaunay triangulations are described in Appendix S3. The results are essentially the same as those of the Series 2 simulations.

### Does the Mantel test capture the spatial variation in response data?

Researchers who use Mantel tests in spatial analysis often assume that the Mantel correlation of a response **D** matrix on a geographic **D** matrix captures the spatial structure that may be present in the response data and, consequently, that regressing response **D** on geographic **D** removes to a large extent the spatial structure from the response **D**, producing residuals without spatial correlation. Appendix S4 shows that this is not the case through a proof-by-example based upon simulated data.

## Discussion

This study has shown that there are more implicit assumptions behind the apparently simple decision to run a Mantel test in the context of spatial analysis than meets the eye.

We provided detailed reasons why the Mantel test is inappropriate to study spatial relationships in response data and supported them with numerical simulation results. The reasons invoked are as follows: (i) the hypothesis of correlation tests of significance that concern raw data differs from that concerning dissimilarity matrices; (ii) the statistics involved in the two types of analyses differ and cannot be reduced to one another; (iii) the Mantel test assumes linearity (or monotonicity) and homoscedasticity in the **D-D** comparison plots and that is not the case except in extreme cases where the range of spatial correlation is equal to or larger than the size of the study area.

Furthermore, our simulation results showed the following:

1. When the range of the variogram controlling the degree of spatial autocorrelation was larger than the interval between sampled grid points, dbMEM analysis was always far more powerful than the Mantel test (Fig. 2a).
2. The Mantel  $R_M^2$  cannot be interpreted as an estimate of the  $R^2$  produced by an analysis of the original response data.
3. In simulations involving truncated distance matrices and Delaunay graph distance matrices, dbMEM analysis was always more powerful than the Mantel test for different values of the truncation distance, except when the truncation value was equal to the interval between the sampled grid points, which created a single distance class with value different from the largest distance in the data set. When more of the distances were kept (i.e. not truncated) in the geographic matrix, the Mantel test had less power to detect SA in the response data.

4. Simulations with a truncation value of 5 were equivalent to a test of the first distance class in a Mantel correlogram; it simply indicated the presence of significant SA in the first distance class. However, when researchers use the Mantel test with truncated distance matrices, they have a specific ecological or genetic dispersion model in mind and they do not truncate to keep only the first distance class.

5. Previous simulations in Legendre, Borcard & Peres-Neto (2005) had shown that spatial variation was, at best, weakly captured by direct regression of a response dissimilarity matrix on a geographic distance matrix. In the present study, we went further and showed that regression on a geographic distance matrix does not control for the spatial structure from response data, and does not produce residuals without spatial correlation.

#### SHOULD THE MANTEL TEST BE USED IN SPATIAL ANALYSIS?

Our conclusions and recommendations to users for spatial analysis of ecological and genetic data are the following.

1. dbMEM analysis by regression or RDA is a more powerful and informative method of spatial analysis than Mantel tests conducted with distance matrices (truncated or not) or Delaunay triangulations. For one, the tests of significance in dbMEM analysis have much greater power to detect SA in data than Mantel tests. Secondly, dbMEM analysis is a method for modelling the spatial structure in univariate or multivariate response data at different scales; the fitted values of the regression or RDA models can be mapped, providing a visual representation of the structure at different spatial scales; the  $R^2_{adj}$  of univariate or multivariate models are unbiased estimates of the portion of the information of the response data explained by the eigenfunctions (Peres-Neto *et al.* 2006). Groups of eigenfunctions representing the variation at different spatial scales can be used in variation partitioning together with other matrices of explanatory variables.

2. Series 2 and 3 simulations showed that when the geographic distance matrix or the Delaunay triangulation are truncated and become binary, the Mantel test becomes identical to a test of the first distance class in a Mantel correlogram and that test has the same power as the test of significance in dbMEM analysis to detect spatial autocorrelation (SA) in response data. The simulation study of Borcard & Legendre (2012) had shown that the Mantel test, used in the context of the Mantel correlogram, had good power to detect SA in data. By opposition, the present series of simulations showed that the ordinary Mantel test has little power to detect SA in data, except in the particular case where a single distance class is studied.

In studies of empirical data, scientists do not know the range of action of SA in the response data. They can use Mantel correlogram analysis to discover it.

3. If ecologists want to use statistical tests to identify SA in field data whose spatial relationships are represented by a Delaunay triangulations or some other connection network, they should check the theoretical framework of their study and decide whether they expect positive or negative SA to be present, and

this for each graph distance. If negative SA is expected for some graph distance, they should use dbMEM or Mantel tests accordingly: for dbMEM, they should use only the eigenfunctions that model negative SA, whereas for Mantel analysis, they should look for significance in the lower tail; these *p*-values are equal to or larger than 0.95 in the output of **vegan**'s `mantel()` function.

To summarize, the Mantel test does not answer the same question and assess the same hypothesis as its raw-data counterparts. When the question concerns the spatial structure of univariate or multivariate data, the lack of concordance of the null hypothesis of the Mantel test with the question produces a test that has low power. In statistics, when several tests of significance are available, one should choose the one that has the highest power, that is, the highest capacity to detect an effect when one is present. The low power of the Mantel test is a symptom of its inadequacy. One should prefer a method with high power, such as dbMEM analysis, to detect spatial structures in data.

#### PARTIAL MANTEL TEST

In ecology and genetics, many papers used partial Mantel tests to control for spatial structures in the analysis of the relationships between response and environmental data, using a geographic distance matrix as covariable. Oden & Sokal (1992) were the first to demonstrate that partial Mantel tests had inflated type I error rates in analyses of dissimilarity matrices computed from independently autocorrelated data. Guillot & Rousset (2013) repeated the Oden & Sokal study in a more extensive way and came to the same conclusion (Appendix S1). This is likely due to the fact that the partial Mantel test suffers from the same problems as the simple test in the context of spatial analysis: inadequate statistic (Eqn 5), lack of linearity of the relationship, and triangular distribution of the distances.

Users of partial Mantel tests should know that when the question of interest is stated in the world of raw data, the analysis should be performed by partial regression or partial canonical analysis and that these linear forms of partial analysis offer greater power than partial Mantel tests. This is especially true in spatial analysis, where simulation studies have shown that the partial Mantel test is less powerful than partial canonical analysis (Legendre, Borcard & Peres-Neto 2005) and can lead to erroneous conclusions (Oden & Sokal 1992).

#### SHOULD THE MANTEL TEST BE USED AT ALL?

Mantel tests are valid and useful when applied to the study of relationships among dissimilarities in dissimilarity matrices. Such questions are rarely encountered in ecology and genetics, but they exist; one example is found in Le Boulengé *et al.* (1996). Mantel tests should simply not be used to test hypotheses that concern the raw data from which dissimilarity matrices can be computed or to control for spatial structures in tests of relationships between two autocorrelated data sets.

In population genetics, researchers often use the Mantel method to test hypotheses of isolation by distance (IBD). What



is the most appropriate and powerful method to test this hypothesis should be the subject of a separate study. It seems clear, however, that a Mantel correlogram or a multivariate variogram would provide more complete and interesting results than a Mantel test because these analyses would indicate what is the range of the autocorrelation in the data. On the other hand, a dbMEM analysis could be conducted to detect and model the spatial correlation in the genetic data. This is done by computing principal coordinates from the genetic distance matrix and using them as response data in a dbMEM analysis by RDA. After running these analyses, researchers could decide what sets of results are the most useful to answer their landscape genetic question.

## Acknowledgements

The authors dedicate this study to the memory of Professor Robert R. Sokal who passed away on 9 April 2012. PL and MJF learned about Mantel and partial Mantel tests in Professor Sokal's laboratory. They also want to thank two anonymous reviewers for stimulating comments. This research was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) research grants to P. Legendre and M.-J. Fortin.

## Data accessibility

This study uses simulated spatially autocorrelated data. The R software used for the simulations presented in section 'Simulations involving spatially autocorrelated data: comparison of Mantel test and dbMEM analysis' of the article is shown in Appendix S5.

## References

- Anselin, L. (1995) Local Indicators of Spatial Association–LISA. *Geographical Analysis*, **27**, 93–115.
- Borcard, D. & Legendre, P. (2002) All-scale spatial analysis of ecological data by means of principal coordinates of neighbour matrices. *Ecological Modelling*, **153**, 51–68.
- Borcard, D. & Legendre, P. (2012) Is the Mantel correlogram powerful enough to be useful in ecological analysis? A simulation study. *Ecology*, **93**, 1473–1481.
- Borcard, D., Legendre, P., Avois-Jacquet, C. & Tuomisto, H. (2004) Dissecting the spatial structure of ecological data at multiple scales. *Ecology*, **85**, 1826–1832.
- Cushman, S.A., McKelvey, K.S., Hayden, J. & Schwartz, M.K. (2006) Gene flow in complex landscapes: testing multiple hypotheses with causal modeling. *American Naturalist*, **168**, 486–499.
- Dietz, E.J. (1983) Permutation tests for association between two distance matrices. *Systematic Zoology*, **32**, 21–26.
- Dray, S., Legendre, P. & Peres-Neto, P.R. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modelling*, **196**, 483–493.
- Dray, S., Pélessier, R., Couteron, P., Fortin, M.-J., Legendre, P., Peres-Neto, P.R. *et al.* (2012) Community ecology in the age of multivariate multiscale spatial analysis. *Ecological Monographs*, **82**, 257–275.
- Dyer, R.J. & Nason, J.D. (2004) Population graphs: the graph theoretic shape of genetic structure. *Molecular Ecology*, **13**, 1713–1727.
- Fortuna, M.A., Albaladejo, R.G., Fernández, L., Aparicio, A. & Bascompte, J. (2009) Networks of spatial genetic variation across species. *Proceedings of the National Academy of Sciences USA*, **106**, 19044–19049.
- Guillot, G. & Rousset, F. (2013) Dismantling the Mantel tests. *Methods in Ecology and Evolution*, **4**, 336–344.
- Le Boulengé, E., Legendre, P., de le Court, C., Le Boulengé-Nguyen, P. & Lan-guy, M. (1996) Microgeographic morphological differentiation in muskrats. *Journal of Mammalogy*, **77**, 684–701.
- Legendre, P. (2000) Comparison of permutation methods for the partial correlation and partial Mantel tests. *Journal of Statistical Computation and Simulation*, **67**, 37–73.
- Legendre, P., Borcard, D. & Peres-Neto, P.R. (2005) Analyzing beta diversity: partitioning the spatial variation of community composition data. *Ecological Monographs*, **75**, 435–450.
- Legendre, P. & Fortin, M.-J. (1989) Spatial pattern and ecological analysis. *Vegetatio*, **80**, 107–138.
- Legendre, P. & Fortin, M.-J. (2010) Comparison of the Mantel test and alternative approaches for detecting complex multivariate relationships in the spatial analysis of genetic data. *Molecular Ecology Resources*, **10**, 831–844.
- Legendre, P. & Legendre, L. (2012) *Numerical Ecology*, 3rd English edn. Elsevier Science BV, Amsterdam.
- Legendre, P., Oksanen, J. & ter Braak, C.J.F. (2011) Testing the significance of canonical axes in redundancy analysis. *Methods in Ecology and Evolution*, **2**, 269–277.
- Legendre, P. & Troussellier, M. (1988) Aquatic heterotrophic bacteria: modeling in the presence of spatial autocorrelation. *Limnology and Oceanography*, **33**, 1055–1067.
- Legendre, P., Borcard, D., Blanchet, F.G. & Dray, S. 2012. *PCNM: MEM spatial eigenfunction and principal coordinate analyses*. R package version 2.1-2. [https://r-forge.r-project.org/R/?group\\_id=195](https://r-forge.r-project.org/R/?group_id=195).
- Manly, B.F.J. (1986) Randomization and regression methods for testing for associations with geographical, environmental and biological distances between populations. *Researches on Population Ecology*, **28**, 201–218.
- Mantel, N. (1967) The detection of disease clustering and a generalized regression approach. *Cancer Research*, **27**, 209–220.
- Mantel, N. & Valand, R.S. (1970) A technique of nonparametric multivariate analysis. *Biometrics*, **26**, 547–558.
- Murphy, M.A., Dezzani, R., Pilliod, D.S. & Storfer, A. (2010) Landscape genetics of high mountain frog populations. *Molecular Ecology*, **19**, 3634–3649.
- Oden, N.L. & Sokal, R.R. (1986) Directional autocorrelation: an extension of spatial correlograms to two dimensions. *Systematic Zoology*, **35**, 608–617.
- Oden, N.L. & Sokal, R.R. (1992) An investigation of three-matrix permutation tests. *Journal of Classification*, **9**, 275–290.
- Oksanen, J., Blanchet, G., Kindt, R., Legendre, P., Minchin, P.R., O'Hara, R.B. *et al.* (2013) *vegan: Community ecology package*. R package version 2.0-10. <http://cran.r-project.org/package=vegan>.
- Peres-Neto, P.R. & Legendre, P. (2010) Estimating and controlling for spatial structure in the study of ecological communities. *Global Ecology and Biogeography*, **19**, 174–184.
- Peres-Neto, P.R., Legendre, P., Dray, S. & Borcard, D. (2006) Variation partitioning of species data matrices: estimation and comparison of fractions. *Ecology*, **87**, 2614–2625.
- Rao, C.R. (1964) The use and interpretation of principal component analysis in applied research. *Sankhyā, Series A*, **26**, 329–358.
- Schlather, M., Malinowski, A., Oesting, M., Boecker, D., Strokorb, K., Engelke, S. *et al.* (2014) *RandomFields: Simulation and analysis of random fields*. R package version 3.0.10. <http://cran.r-project.org/package=RandomFields>.
- Sokal, R.R. (1979) Testing statistical significance of geographic variation patterns. *Systematic Zoology*, **28**, 227–232.
- Sokal, R.R. (1986) Spatial data analysis and historical processes. *Data analysis and Informatics, IV* (eds E. Diday *et al.*), pp. 29–43. North-Holland, Amsterdam.
- Sokal, R.R. & Rohlf, F.J. (1995) *Biometry – The Principles and Practice of Statistics in Biological Research*, 3rd edn. W.H. Freeman and Co., New York.

Received 24 February 2015; accepted 8 June 2015

Handling Editor: Pedro Peres-Neto

## Supporting Information

Additional Supporting Information may be found in the online version of this article.

**Appendix S1.** Analysis of spatially correlated data and Mantel test: who has shown what?.

**Appendix S2.** Analysis of simulated random autocorrelated surfaces.

**Appendix S3.** Series 3 simulations involving Delaunay triangulations.

**Appendix S4.** Regression on a geographic distance matrix does not control for SA in data.

**Appendix S5.** Software used in the simulations.

## Appendices to:

Legendre, P., Fortin, M.-J. & Borcard, D. (2015) Should the Mantel test be used in spatial analysis? *Methods in Ecology & Evolution*, **6**, 1239–1247. doi: 10.1111/2041-210X.12425

## Appendix S1

### ANALYSIS OF SPATIALLY CORRELATED DATA AND MANTEL TEST: WHO HAS SHOWN WHAT?

This section reviews some of the papers that showed important characteristics of tests of significance in the presence of spatial correlation, including Mantel and partial Mantel tests. Simulation studies previously published in the ecological and statistical literatures have touched upon different aspects of the problem of analysing the correlation between spatially correlated data. This brief review of the literature puts our simulation results in the context of the results that are already known about the Mantel and partial Mantel tests and show the significance of our new findings in that context.

Spatial autocorrelation is often taken to mean any kind of spatial dependence (or spatial structure) in data. More formally, one can distinguish *induced spatial dependence*, which is the type of spatial dependence present in a variable due to the causal influence of an explanatory variable that is spatially structured, and *spatial autocorrelation in the strict sense* which is the spatial dependence that remains in the residuals after the effects of all pertinent explanatory variables have been taken into account (Legendre & Legendre 2012, Section 1.1).

- To our knowledge, Bivand (1980) was the first to publish a short series of simulation results showing that the test of significance of the correlation coefficient between two variables that were uncorrelated to each other but strongly spatially autocorrelated had inflated levels of type I error, meaning that the test rejected the null hypothesis more often than predicted by the  $\alpha$  significance level. He also showed that when only one of the variables was autocorrelated, there was no such strong effect. In their well-cited book on spatial processes, Cliff & Ord (1981) included a figure (their Fig. 7.2) presenting some of the Bivand simulation results.

- Manly (1986) simulated pairs of spatially correlated multivariate data on points on a map. The data were then transformed into distance matrices **A** and **B** and tested for significance by regressing **A** on **B**; a third distance matrix **C** representing geographic distances was included in the regression equation. For the test, the statistic was the regression coefficient  $b_i$ , which was a partial regression coefficient since the **C** distances were also in the regression equation. The test of significance involved Mantel-like permutations of matrix **A**. The simulation results did not demonstrate any major effect of spatial correlation on the estimated values of the regression coefficients in simple ( $\mathbf{A} \sim \mathbf{B}$ ) and multiple regression ( $\mathbf{A} \sim \mathbf{B} + \mathbf{C}$ ) on distance matrices. This may be due in part to the use of a suboptimal test statistic in the Manly (1986) paper. In the 1997 edition of his book (p. 180), Manly modified the testing procedure, recommending to use the pivotal statistic  $t_i = b_i/\text{SE}(b_i)$  where  $\text{SE}(b_i)$  is the standard error of  $b_i$ , instead of  $b_i$  as the test statistic in the permutation test.

- Smouse *et al.* (1986) suggested two ways of testing a partial Mantel statistic  $r_M(\mathbf{AB.C})$ . Their first method was the same as the original regression method of Manly (previous point). The second was to compute the residuals of the partial correlations of **A** on **C** and of **B** on **C**, then

carry out a simple Mantel test between the two residual matrices. Later, Oden & Sokal (1992) used numerical simulations to compare three methods of partial Mantel analysis in the situation where two data sets were spatially autocorrelated, but not correlated to each other. They devised two ways of simulating spatially autocorrelated data. The first one implemented the isolation-by-distance model of population genetics, the second consisted in simulating spatial autocorrelation in matrices **A** and **B** using a spatial covariance matrix that specified the relationships among the points, which were exponentially declining with geographic distance. The simulations showed that the type I error rates of the three partial Mantel testing procedures were inflated in the presence of strong spatial autocorrelation.

- Legendre (2000) used simulations to compare the type I error rates and powers of four permutation methods used for testing the correlation among distance matrices in partial Mantel tests. His simulations did not involve spatially autocorrelated data. The other studies that involved spatially autocorrelated data had simulated vectors or matrices of spatially autocorrelated data, which were then turned into distance matrices and analysed with a matrix **C** of geographic relationships as covariables. In the Legendre (2000) simulations on the contrary, three distance matrices were independently generated and were then correlated to one another using a correlation model. In that way, the tests of significance were really about correlations *between the distance matrices*, on which the null hypothesis of the Mantel test is based; the Mantel test does not test a hypothesis about correlation in the raw data. The simulation results showed that three of the permutation methods under study were appropriate, to the exclusion of method 3 (which was the second method of Smouse *et al.*, 1986).
- Following a NCEAS working group<sup>1</sup>, Legendre *et al.* (2002) published extensive simulation results showing that the test of significance of the Pearson correlation coefficient between two variables that were uncorrelated to each other but strongly spatially correlated (due either to induced spatial dependence or to true spatial autocorrelation) had inflated levels of type I error, which made the test invalid, and that this effect disappeared when only one of the variables was spatially correlated. These results confirmed the less extensive simulation results of Bivand (1980). They also showed that Dutilleul's (1993) modified *t*-test for the correlation coefficient, which takes the spatial correlation of the variables into account, effectively corrected for the spatial correlation in the data and produced results with correct levels of type I error.
- Castellano & Balletto (2002) used simulations to show that the type I error of the partial Mantel test was correct. These authors used incorrect partial testing procedures, so their conclusions remain doubtful. That paper started an exchange in the literature, which is discussed in Appendix 3 of the Legendre & Fortin (2010) paper.
- In 2005, Legendre *et al.* published a paper comparing canonical redundancy analysis (RDA) to Mantel tests for the analysis of simulated multivariate, spatially structured data. The questions were to determine (1) which method had the highest power to detect a relationship between the two data sets when they were spatially autocorrelated and (2) which method had the highest power to detect spatial structures in the data. The two data sets represented community

---

<sup>1</sup> Working Group "Integrating the Statistical Modeling of Spatial Data in Ecology", 1999-2000, supported by the National Center for Ecological Analysis and Synthesis (NCEAS), a Center funded by NSF (Grant # DEB-94-21535), the University of California at Santa Barbara, and the State of California, USA.

composition data and environmental variables. The simulation results showed that partial RDA had much higher power than partial Mantel tests (1) to detect relationships between the two data sets when they were related and (2) to detect the presence of spatial structures in the species (response) data. In a follow-up paper (2008), additional simulations showed identical results for community composition data generated using Hubbell's neutral model. The main conclusion of these papers was that the Mantel test is inappropriate to test hypotheses concerning correlations in raw data; its use should be restricted to the study of correlations between the distances in distance matrices. – These papers did not study the combination that was of interest in the Guillot & Rousset (2013) paper, i.e. two unrelated data sets that were both spatially autocorrelated. The 2005 paper showed, however, that simple RDA and the simple Mantel test had correct levels of type I error when the two data sets were unrelated to each other and one of them was spatially autocorrelated.

- The Legendre & Fortin (2010) paper was concerned with the relative powers of the Pearson and Mantel correlations ( $r$  and  $r_M$ ) for the study of genetic data. Preliminary simulations that did not involve spatially correlated data showed (again) that both methods had correct levels of type I error. More importantly, the simulations showed that in tests of significance of the relationship between simple variables and multivariate data tables, the power of linear correlation, regression and canonical analysis was far greater than that of the Mantel test and derived forms, meaning that the former methods are much more likely than the latter to detect a relationship when one is present in the data. Examples of difference in power are given for the detection of spatial gradients. Furthermore, the Mantel test does not correctly estimate the proportion of the original data variation explained by spatial structures. The Mantel test should not be used as a general method for the investigation of linear relationships or spatial structures in univariate or multivariate data. Its use should be restricted to tests of hypotheses that can only be formulated in terms of distances. An example of a study where the hypotheses clearly and only involved distances is Le Boulengé *et al.* (1996).

- Finally, using simulations, Guillot & Rousset (2013) found that simple Mantel tests between two autocorrelated variables that were not correlated to each other had inflated rates of type I error, a result in the same line as those obtained for Pearson correlations by Bivand (1980) and by Legendre *et al.* (2002). That result is expected for two vectors or matrices of raw data that are not independent and identically distributed (abbreviated *i.i.d.*). For the same kind of data, they also found that partial Mantel tests did not adequately correct for the presence of spatial structures, so that the tests also had inflated type I error rates. The simulations carried out by Guillot & Rousset (*ibid.*) for partial Mantel tests are more detailed than those of Oden & Sokal (1992) and they confirm their conclusions. Guillot & Rousset must be commended for their effort, but one would have expected them to state that their conclusion had been published 21 years before, instead of claiming that result as their own.

Except for the Legendre (2000) paper, all the above-mentioned studies, including the Guillot & Rousset (2013) paper, simulated *raw data* that were spatially autocorrelated; they were then transformed into distance matrices for Mantel testing. Admittedly, that corresponds to the way most researchers use partial Mantel tests to assess the relationship between data sets while controlling the type I error rate inflation due to spatial autocorrelation. However, and as mentioned above, the partial Mantel test does not test the correlation between two data sets, but between the distances in two distance matrices. This explains why Oden & Sokal (1992) and Guillot & Rousset (2013) found that it did not correct adequately for spatial correlation in data



and had inflated type I error rates. The fault is not with the partial Mantel test but with the inadequacy of the data that were analysed using that method. In the Legendre (2000) simulations, on the contrary, the partial Mantel test was found to have correct type I error rates because the data that were subjected to it were distances that were intercorrelated (although not spatially correlated), and so they corresponded to the null hypothesis of the test.

## References

- Bivand, R. (1980) A Monte Carlo study of correlation coefficient estimation with spatially autocorrelated observations. – *Quaestiones Geographicae*, **6**, 5–10.
- Castellano S, Balletto E (2002) Is the partial Mantel test inadequate? *Evolution* **56**, 1871–1873.
- Cliff, A. D. & Ord, J. K. (1981) *Spatial processes – Models and applications*. Pion, London.
- Dutilleul, P. (1993) Modifying the *t* test for assessing the correlation between two spatial processes. *Biometrics*, **49**, 305–314.
- Guillot, G. & Rousset, F. (2013) Dismantling the Mantel tests. *Methods in Ecology and Evolution*, **4**, 336–344. doi: 10.1111/2041-210x.12018.
- Le Boulengé, É., Legendre, P., de le Court, C., Le Boulengé-Nguyen, P. & Languy, M. (1996) Microgeographic morphological differentiation in muskrats. *Journal of Mammalogy*, **77**, 684–701.
- Legendre, P. (2000) Comparison of permutation methods for the partial correlation and partial Mantel tests. *Journal of Statistical Computation and Simulation*, **67**, 37–73.
- Legendre, P., Borcard, D. & Peres-Neto, P. R. (2005) Analyzing beta diversity: partitioning the spatial variation of community composition data. *Ecological Monographs*, **75**, 435–450.
- Legendre, P., Borcard, D. & Peres-Neto, P. R. (2008) Analyzing or explaining beta diversity: Comment. *Ecology*, **89**, 3238–3244.
- Legendre, P., Dale, M. R. T., Fortin, M.-J., Gurevitch, J., Hohn, M. & Myers, D. (2002) The consequences of spatial structure for the design and analysis of ecological field surveys. *Ecography*, **25**, 601–615.
- Legendre, P. & Fortin, M.-J. (2010) Comparison of the Mantel test and alternative approaches for detecting complex multivariate relationships in the spatial analysis of genetic data. *Molecular Ecology Resources*, **10**, 831–844.
- Legendre, P. & Legendre, L. (2012) *Numerical ecology. 3rd English edition*. Elsevier Science BV, Amsterdam.
- Manly, B. F. J. (1986) Randomization and regression methods for testing for associations with geographical, environmental and biological distances between populations. *Researches on Population Ecology*, **28**, 201–218.
- Manly, B. F. J. (1997) *Randomization, bootstrap and Monte Carlo methods in biology. 2nd edition*. Chapman and Hall, London.

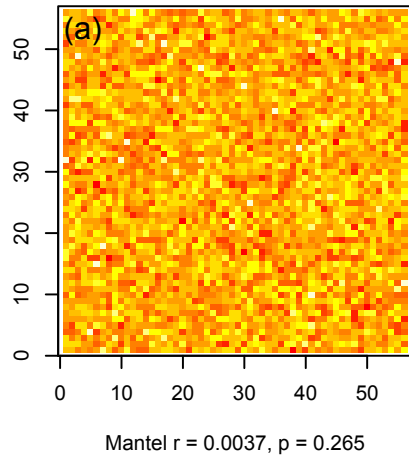
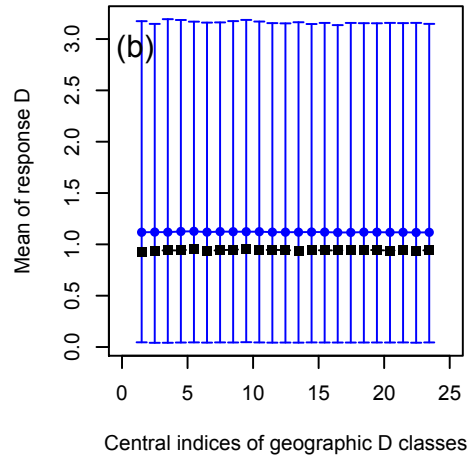
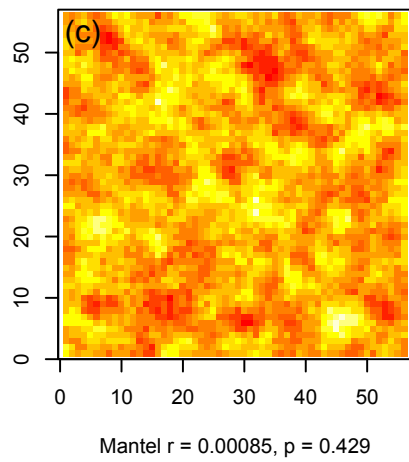
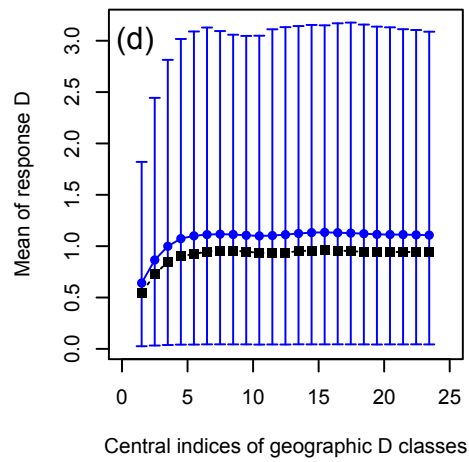
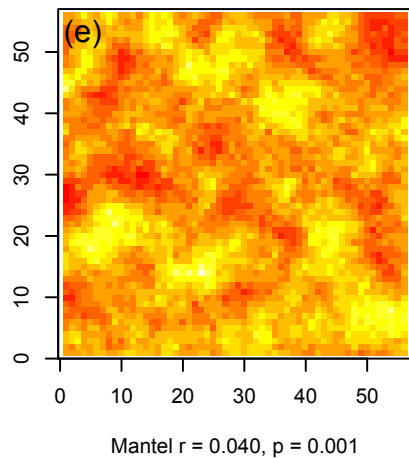
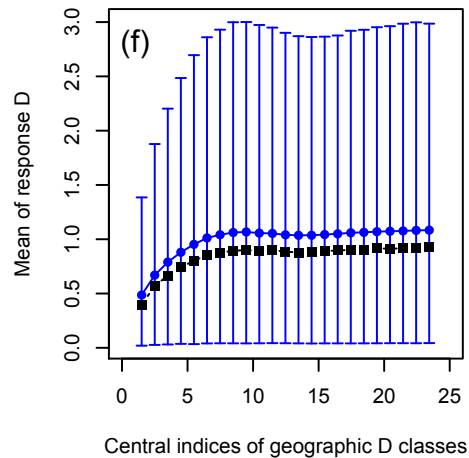
Oden, N. L. & Sokal, R. R. (1992) An investigation of three-matrix permutation tests. *Journal of Classification*, **9**, 275–290.

Smouse, P. E., J. C. Long & R. R. Sokal. 1986. Multiple regression and correlation extensions of the Mantel test of matrix correspondence. *Syst. Zool.* 35: 627–632.

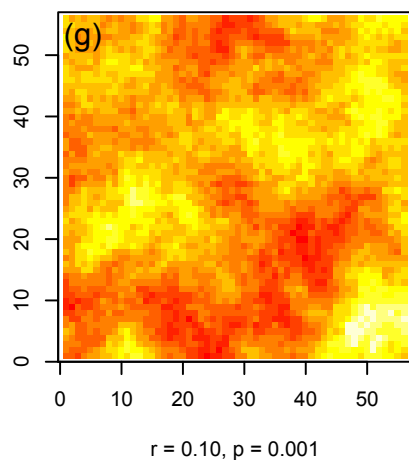
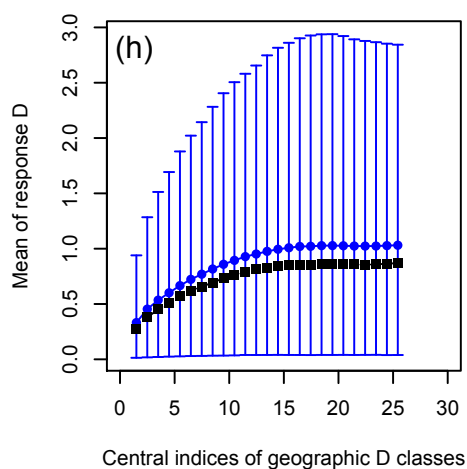
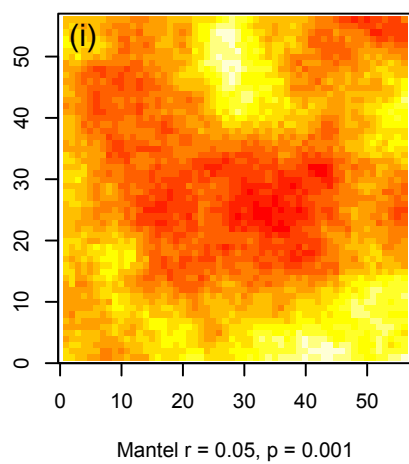
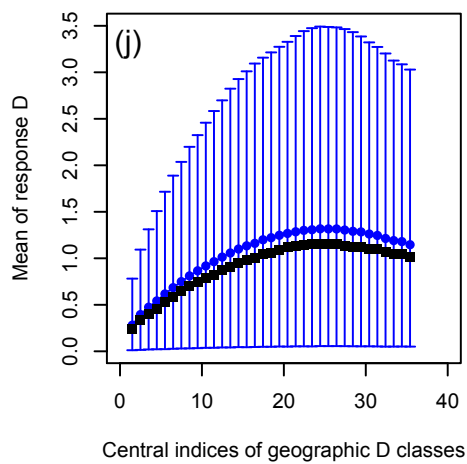
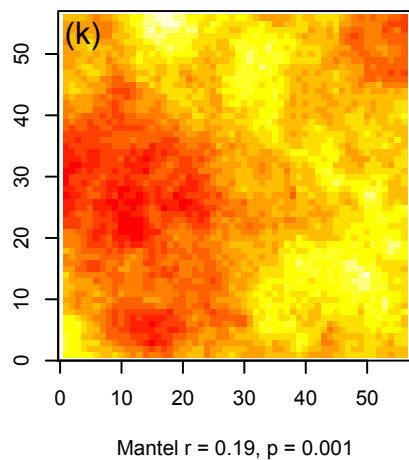
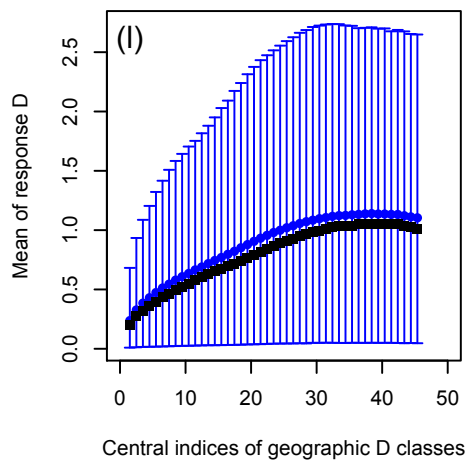
## **Appendix S2**

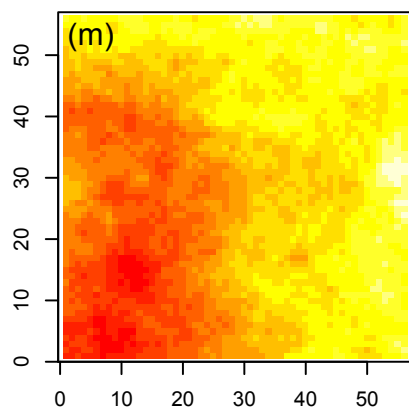
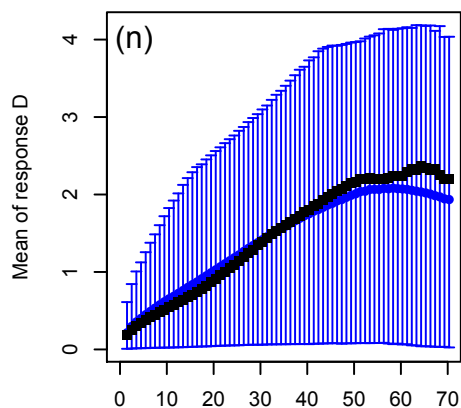
### **ANALYSIS OF SIMULATED RANDOM AUTOCORRELATED SURFACES**

Random autocorrelated variables were generated by random Gaussian field simulations on a ( $56 \times 56$  pixel) grid with autocorrelation controlled by a spherical variogram with range values of  $\{0, 5, 10, 20, 30, 40, 50, 60, 70\}$  pixel units. The values at 100 points forming a regular grid were sampled from each surface, with horizontal and vertical spacing of 5 pixel units. Examples of simulated surfaces are shown and analysed in Fig. S2.1 of this appendix.

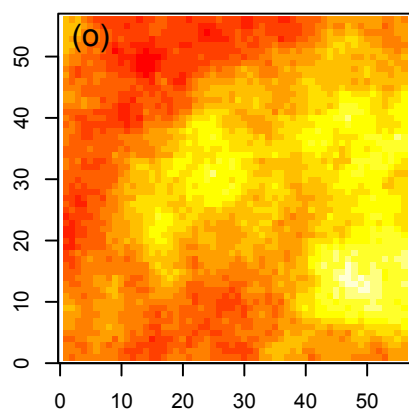
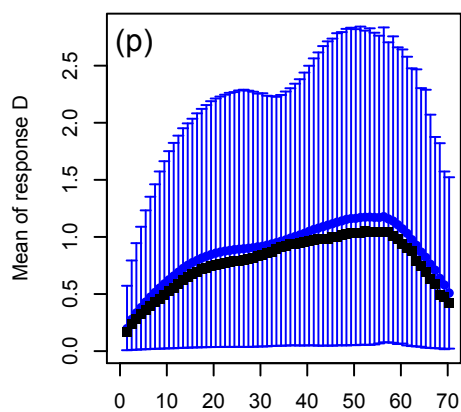
**Response surface, Range=0****Mean response.D vs geo.D classes****Response surface, Range=5****Mean response.D vs geo.D classes****Response surface, Range=10****Mean response.D vs geo.D classes**



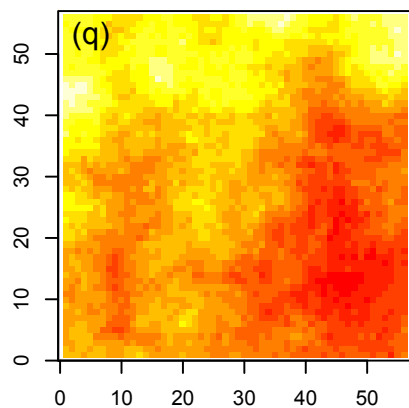
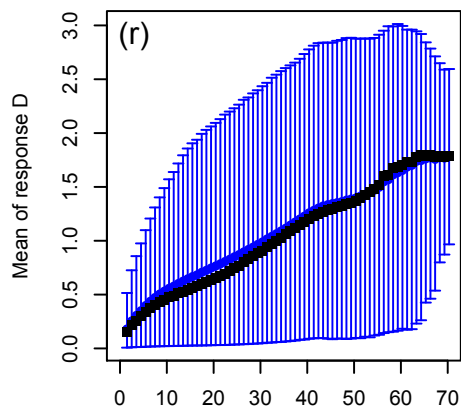
**Response surface, Range=20****Mean response.D vs geo.D classes****Response surface, Range=30****Mean response.D vs geo.D classes****Response surface, Range=40****Mean response.D vs geo.D classes**

**Response surface, Range=50**Mantel  $r = 0.47$ ,  $p = 0.001$ **Mean response.D vs geo.D classes**

Central indices of geographic D classes

**Response surface, Range=60**Mantel  $r = 0.26$ ,  $p = 0.001$ **Mean response.D vs geo.D classes**

Central indices of geographic D classes

**Response surface, Range=70**Mantel  $r = 0.45$ ,  $p = 0.001$ **Mean response.D vs geo.D classes**

Central indices of geographic D classes

**Fig. S2.1.** For each value of variogram range (Range = 0 to 70), one of the possible simulated surfaces is shown on the left, using a colour scale from dark red (low values) to pale yellow (high values). Right: a plot similar to Fig. 1c of the main paper is presented, where the means (circles) and medians (squares) of the dissimilarities computed from the simulated values of the response variable are shown as a function of the geographic distance classes. Each mean value is accompanied by its empirical 95% coverage interval. Mantel  $r$  and  $p$ -values of the **D-D** comparisons are shown underneath the response surface maps.

## Appendix S3

### SERIES 3 SIMULATIONS INVOLVING DELAUNAY TRIANGULATIONS

The purpose of the third series of simulations was to reproduce circumstances where the number of pairwise comparisons among sites is reduced to reflect species predation avoidance behaviour or limited dispersal abilities in fragmented landscapes (Braunisch *et al.* 2010). In such cases, species move through fragmented landscapes using nearby patches in a stepping-stone fashion instead of moving across inhospitable areas over long distances. A reduced number of pairwise steps among sites can also occur in translocation experiment field studies (Bélisle *et al.* 2001); likewise, in studies of invasive species that have not invaded the whole study area or the spread of genetically modified organisms (GMO) that hybridize with wild or unmodified cultivated forms in contact zones. To represent a limited number of edges linking sites, different types of networks have been developed, going from the minimum spanning tree (Gower & Ross 1969; Urban & Keitt 2001), to the more complex relative neighbourhood graph (Toussaint 1980), Gabriel graph (Gabriel & Sokal 1969; Naujokaitis-Lewis *et al.* 2013), and finally the Delaunay triangulation (Dirichlet 1850) and its landscape equivalent, the minimum planar graph (Fall *et al.* 2007). The first four form a nested series of networks with increasing connectedness (Toussaint 1980). By far the most commonly used algorithm in landscape ecology and genetics is the Delaunay network (e.g. Goldberg & Waits 2010; Koen *et al.* 2012) to represent the stepping-stone behaviour of species (Saura *et al.* 2014). These Delaunay networks can then be truncated to match the species dispersal ability.

So we carried out a third series of simulations to study the power of truncated Delaunay networks. The edges of a Delaunay triangulation represent the spatial relationships between points. For each simulated surface, a random sample of  $N = 50$  points was selected from the regular grid of 100 points, and a Delaunay triangulation and a matrix of dbMEM eigenfunctions were computed for the selected points.

Then, a “graph distance matrix” was computed along the edges of each Delaunay triangulation using function `delaunay.distance()` of the **spatstat** package (Baddeley *et al.* 2014). The distance was *the number of edges* along the shortest path between any pair of points in the connection network. In studies as those described above, researchers may choose to truncate the distance matrix, replacing any value larger than a selected threshold by a large distance, which can be the largest distance actually found in the matrix `max.D` (that was the case in our simulations), or by some larger distance chosen by the user. The following values of truncation threshold were used:  $thresh = \{1, 3, 5, 10\}$ . Since no graph distance was ever larger than 10 in our simulations, the simulations with  $thresh = 10$  used the full graph distance matrix without any truncation. The autocorrelated data sets were analysed with respect to geography through a dbMEM analysis using the full set of eigenfunctions modelling positive spatial correlation, as in series 1 simulations, and a Mantel test using the truncated Delaunay “graph distance matrix”.

Fig. S3.1 shows the mean rejection rates obtained in the various simulations, for different amounts of SA (variogram ranges, above) and different truncation thresholds ( $thresh = \{1, 3, 5, 10\}$ ). The following observations can be made:

- For Delaunay graph distance matrices truncated at distance 1, the results of the Mantel test had the same rejection rates as the dbMEM method for all amounts of SA in the response data. In



these matrices, only two distance classes are present: 1 and max.D. The latter was the largest value encountered in the Delaunay graph distance matrix; it varied from 7 to 10 in our simulations, depending on the selected subset of 50 points. This form of Mantel test is the same as that conducted for the first distance class in a Mantel correlogram. This equivalence of the two tests was encountered in the series 2 simulation results. More about this in the Discussion of the main paper.

- For truncation levels larger than 1 (i.e.  $thresh = \{3, 5, 10\}$ ), the power of the Mantel test was always much lower than that of the dbMEM test, except when no spatial autocorrelation was present in the data, which occurred when the range of the variogram was smaller than (range = 0) or equal to (range = 5) the spacing between the points of the regular grid; that spacing determined the closest possible spacing of pairs of points in the Delaunay triangulation.

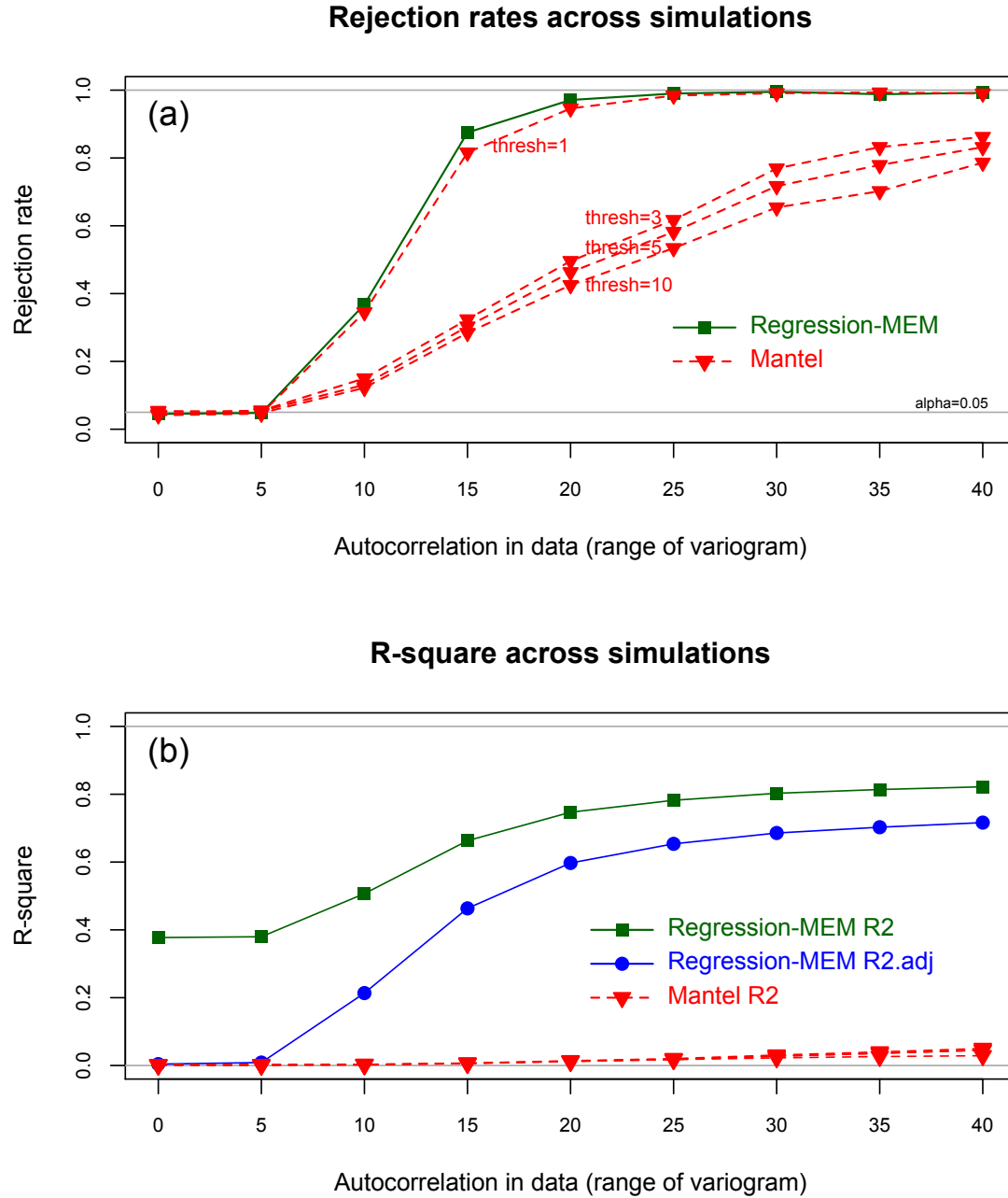
The point where the power of the dbMEM test comes near 1 in Fig. S3.1a (i.e. when range = 20) cannot be compared to that same point in Figs. 2a and 3a (where power is nearly 1 at range = 10) because 100 data points were used in Figs. 2a and 3a whereas Fig. S3.1a used only 50 points.

Fig. S3.1b shows that the  $R^2$  of the Mantel test is always smaller than the  $R^2$  and  $R^2_{adj}$  of the dbMEM analysis. They are both equal to 0 when there is no spatial structure in the data, i.e. when the range of the variogram was 0 or 5 in our simulations. This simply illustrates the well-established fact that the  $R^2$  of regression or canonical analysis is unrelated to the  $R^2$  of the Mantel test, as discussed in a previous paper (Legendre & Fortin 2010).

## References

- Baddeley, A., Turner, R. & Rubak, E. (2014) *spatstat: Spatial point pattern analysis, model-fitting, simulation, tests*. R package version 1.38-1. <http://cran.r-project.org/package=spatstat>.
- Bélisle, M., Desrochers, A. & Fortin, M.-J. (2001) Landscape barriers to forest birds: a test with homing experiments. *Ecology*, **82**, 1893–1904.
- Braunisch, V., Segelbacher, G. & Hirzel, A. (2010) Modelling functional landscape connectivity from genetic population structure: a new spatially explicit approach. *Molecular Ecology*, **19**, 3664–3678.
- Dirichlet, G.L. (1850) Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Journal für die reine und angewandte Mathematik*, **40**, 209–234.
- Fall, A., Fortin, M.-J., Manseau, M. & O'Brien, D. (2007) Spatial Graphs: Principles and Applications for Habitat Connectivity. *Ecosystems*, **10**, 448–461.
- Gabriel, K.R. & Sokal, R.R. (1969) A new statistical approach to geographic variation analysis. *Systematic Zoology*, **18**, 259–278.
- Goldberg, C.S. & Waits, L.P. (2010) Comparative landscape genetics of two pond-breeding amphibian species in a highly modified agricultural landscape. *Molecular Ecology*, **19**, 3650–3663.

- Gower, J.C. & Ross, G.J.S. (1969) Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, **18**: 54–64.
- Koen, E.L., Bowman, J., Garroway, C.J., Mills, S.C. & Wilson, P.J. (2012). Landscape resistance and American marten gene flow. *Landscape Ecology*, **27**, 29–43.
- Legendre, P. & Fortin, M.-J. (2010) Comparison of the Mantel test and alternative approaches for detecting complex multivariate relationships in the spatial analysis of genetic data. *Molecular Ecology Resources*, **10**, 831–844.
- Naujokaitis-Lewis, I., Rico, Y., Lovell, J., Fortin, M.-J., & Murphy, M. (2013) Implications of incomplete networks on estimation of landscape genetic connectivity. *Conservation Genetics*, **14**, 287–298.
- Saura, S., Bodin, Ö. & Fortin, M.-J. (2014) Stepping stones are crucial for species' long-distance dispersal and range expansion through habitat networks. *Journal of Applied Ecology*, **51**, 171–182.
- Toussaint, G. (1980) The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, **12**: 261–268.
- Urban, D. & Keitt, T. (2001) Landscape connectivity: a graph-theoretic perspective. *Ecology*, **82**, 1205–1218.



**Fig. S3.1.** (a) Rejection rates (i.e. number of rejections of  $H_0$  divided by the number of simulations, which was 1000) of the regression–MEM and Mantel tests as a function of the degree of autocorrelation (variogram range) in the simulated data involving Delaunay triangulations with various truncation levels ( $thresh = \{1, 3, 5, 10\}$ ;  $thresh = 10$  produces no truncation). (b) Median  $R$ -squares of the two methods of analysis. The median adjusted  $R$ -square ( $R^2_{adj}$ ) of the regression–MEM test, which is an unbiased estimate of the explained variation, is also shown. No  $R^2_{adj}$  statistic is available for Mantel tests.

## Appendix S4

### REGRESSION ON A GEOGRAPHIC DISTANCE MATRIX DOES NOT CONTROL FOR SA IN DATA

This Appendix offers a proof-by-example that regression on a geographic distance matrix does not control for the spatial correlation that may be found in response data.

Legendre *et al.* (2005) have already shown that spatial variation is, at best, weakly captured by regressing a response distance (or dissimilarity) matrix on a geographic distance matrix, whatever its form:  $D(XY)$ ,  $D(3^{\text{rd}}$  degree polynomial of  $X$  and  $Y$ ), or  $\ln(D(XY))$ , where  $XY$  is a matrix of geographic coordinates. In this Appendix, we address the other facet of this question, which is: *Does regression on a geographic distance matrix remove the spatial correlation in response data, producing residuals without spatial correlation?*

We will approach this question using example data. We wrote an R function, called `gen.SA.data()`, that readers can use to generate a variable of simulated spatially autocorrelated data over a geographic surface (regular grid). The function is provided at the end of this Appendix. The calculation steps are the following:

1. Check that the data do indeed contain spatially correlated data. To that aim, we apply dbMEM analysis to the response data using multiple linear regression, since there is a single response variable. If the data were multivariate, redundancy analysis (RDA) would be used instead of multiple regression. MEM analysis has been proven capable of identifying spatial correlation in response data (i.e. spatial structures of various kinds, be they the result of a process producing spatial autocorrelation in data, or the result of spatial dependence induced by explanatory variables); this has been demonstrated in Borcard & Legendre (2002), Borcard *et al.* (2004), Dray *et al.* (2006) and Legendre & Legendre (2012, Chapter 14). We make note of the R-square, adjusted R-square and p-value of the MEM analysis as proof of the presence of spatial correlation in the generated data. The function outputs a data matrix ("Surface") from which a colour map of the generated data can be drawn.
2. The function then carries out Mantel tests of the generated response data vs. geographic distances expressed in various forms: raw geographic distances ( $D(XY)$ ), square-rooted distances ( $\sqrt{D(XY)}$ ), and  $\log_e$ -transformed distances ( $\ln(D(XY))$ ). The R-square and p-value of each Mantel analysis is noted.
3. From the output of the function, one can recuperate the generated data and the grid coordinates and compute distance matrices (see the *Example run* below). One can then compute a regression of the response distances on the geographic distances for each form: raw, square-rooted and log distances.
4. We will focus on the fitted values and the residuals of these regressions. The residuals are not really distances (about half of the values are negative), but they can still be tested for the presence of spatial correlation. The residual distances are regressed on the set of MEM eigenfunctions recuperated from the `gen.SA.data()` function. Testing is carried out using the test of significance proposed by McArdle & Anderson (2001), which uses a response distance matrix directly for the calculation. We examine the R-square, adjusted R-square and p-value of the permutation tests of both the fitted distances and the residual distances.



5. If the test of the residual distances is not significant, it means that no significant spatial correlation has been identified in the residual distances. If the test is significant, and especially, if the adjusted R-square is high, we must conclude that spatial correlation is still present in the response data after residualization by regression on the various kinds of geographic distance matrices. If the test is significant but the adjusted R-square is lower than that of the regression of the generated data on MEM eigenfunctions, it means that regression on the geographic distance matrix has controlled the spatial correlation in the response data less efficiently than regression on MEM eigenfunctions.

The function and additional calculations will now be run to provide an example of the analysis. Readers are invited to run the function again (it will generate a different spatially autocorrelated surface every time) and check that the results are similar to those reported here.

### Example run

**1. Load the files** containing the `gen.SA.data()` function, the `dbRDA.D()` function, and the necessary R packages. The list of packages and their locations are given in the *Details* paragraph of the `gen.SA.data()` function documentation.

Functions `gen.SA.data.R` and `dbRDA.D.R` are included in this Appendix, from which they can be copied to text files and loaded to the R console.

**2. Function `gen.SA.data()` generates spatially autocorrelated data** on a surface (square grid) and analyses it using MEM (by multiple linear regression) and Mantel tests.

# Generate a 20×20 spatially autocorrelated (SA) surface (400 points) with variogram range = 5. Only the MEM eigenfunctions modelling positive SA are kept for regression analysis.

```
res = gen.SA.data(nx=20, range=5, var=1, nperm1=999, nperm2=999)
```

```
summary(res)
      Length Class      Mode
R2.dbMEM      2  -none-   numeric
p.dbMEM       2  -none-   numeric
R2.Mantel     3  -none-   numeric
p.Mantel      3  -none-   numeric
Surface     400  -none-   numeric
surf        400  -none-   numeric
grid.coord    2  data.frame list
dbMEM       189  data.frame list
```

```
dim(res$dbMEM)
```

```
# [1] 400 189
```

```
res$R2.dbMEM      # R-square, adjusted R-square of MEM analysis by regression
```

```
# [1] 0.9176085 0.8434561
```

```
res$p.dbMEM      # Parametric and permutational p-values
```

```
# [1] 1.188061e-59 1.000000e-03
```

```
# The presence of significant spatial autocorrelation in the response data is confirmed
```

```
res$R2.Mantel      # Mantel R-square for geographic D, sqrt(D), and ln(D)
```

```
# [1] 0.0008160981 0.0012585326 0.0020933597
```

```
res$p.Mantel      # Mantel p-values for geographic D, sqrt(D), and ln(D)
```

```
# [1] 0.038 0.012 0.003
```

```
# Mantel analysis explains a very small fraction of the response distance matrix variation
```

### 3. Plot a map of the original random autocorrelated surface

```
require(graphics)
```

```
image(1:20, 1:20, t(res$Surface), main="Map of response surface, Range=5", xlim=c(0,21),  
ylim=c(0,21), xlab="", ylab="", asp=1)
```

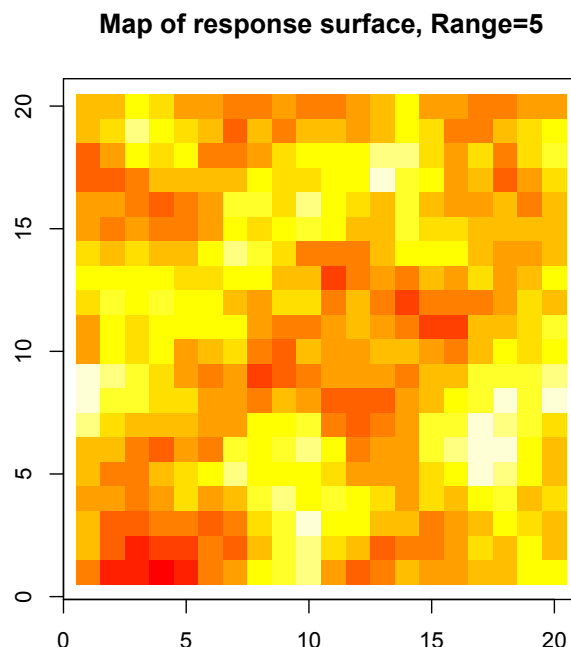


Fig. S4.1. Map of the response surface generated on a 20×20 grid for this example.

### 4. Check that the response data, expressed in distance matrix form, displays significant spatial structure.

This analysis will confirm that the dbRDA.D() function produces the same R-square, adjusted R-square and p-value as regular regression (results are shown above).

```
# Load function dbRDA.D() to compute RDA following McArdle & Anderson (2001), using a  
dissimilarity matrix and a matrix of explanatory variables.
```

```
res0.dbRDA = dbRDA.D(dist(res$surf), res$dbMEM, nperm=999, compute.eig=FALSE)
```

```
# Rank of X centred = 189
```

```
res0.dbRDA$Rsquare          # R-square and adjusted R-square; same results as above
```

```
# [1] 0.9176085 0.8434561
```

```
res0.dbRDA$P.perm          # Permutational p-value
```

```
# [1] 0.001
```

### 5. Compute the fitted and residual distances and test them for spatial correlation

```
# First, compute vectors containing the two distance matrices, unfolded
```

```

Y.D.vec = as.vector(dist(res$surf))           # Response data D, vector length = 79800
XY.D.vec = as.vector(dist(res$grid.coord))    # Geographic D, vector length = 79800
# Regress response distances (Y.D) on simple geographic distances (XY.D) in vector form
res.lm.D = lm(Y.D.vec ~ XY.D.vec)

summary(res.lm.D)$r.squared                    # Same as Mantel R-square above
# [1] 0.0008160981

```

### 5.1. Fitted distances

```

fitted.D = fitted(res.lm.D)                   # Length of the vector: 79800

range(fitted.D)
# [1] 1.039841  1.161839

# Turn the fitted dissimilarities into a matrix with class "dist"
fitted.D.mat <- as.dist(matrix(NA,400,400))
fitted.D.mat[] <- fitted.D

# Do the fitted distances contain SA? dbRDA.D of the fitted distance matrix against MEM.
res.dbRDA.fit = dbRDA.D(fitted.D.mat, res$dbMEM, nperm=999, compute.eig=FALSE)

res.dbRDA.fit$Rsquare                         # Fitted D: R-square and adjusted R-square
# [1] 0.51806244  0.08431864

res.dbRDA.fit$P.perm                         # Permutational p-value
[1] 0.001

# The fitted distances do account for a small albeit significant amount of SA, which was present
in the generated data.

```

### 5.2. Residual distances

```

resid.D = residuals(res.lm.D)                 # Length of the vector: 79800

range(resid.D)
# [1] -1.146697  4.388709

length(which(resid.D < 0)) / (200*399)        # 0.5735589
# In this particular example, 57.4% of the residual "distances" were negative.

hist(resid.D)

# Turn the residual dissimilarities into a matrix with class "dist"
resid.D.mat = as.dist(matrix(NA,400,400))
resid.D.mat[] = resid.D

# Do the residual distances contain SA? dbRDA of the residual distance matrix against MEM.

```

# Load function dbRDA.D() to compute RDA following McArdle & Anderson (2001), using a dissimilarity matrix and a matrix of explanatory variables.

```
res.dbRDA = dbRDA.D(resid.D.mat, res$dbMEM, nperm=999, compute.eig=FALSE)
```

```
summary(res.dbRDA)
```

```
res.dbRDA$Rsquare                                # Residual D: R-square and adjusted R-square
```

```
# [1] 0.7151880 0.4588571
```

```
res.dbRDA$P.perm                                # Permutational p-value
```

```
# [1] 0.001
```

# The residual distances *do* contain significant SA in substantial amount.

Readers are invited to regress the response distances (Y.D) on the square-rooted ( $\sqrt{XY.D}$ ) and log-transformed ( $\ln(XY.D)$ ) geographic distances, compute matrices of fitted and residual distances from the regressions, and test these distance matrices for presence of SA by MEM analysis using the dbRDA.D() function, following the script provided above where Y.D is regressed on XY.D. The detailed steps are not presented to save space. They have been computed, however, and the results are assembled in Table S4.1 for the random autocorrelated response data generated during our run of the gen.SA.data() function. The adjusted R-squares of the dbMEM.D analyses against MEM eigenfunctions are the statistics of interest because they represent unbiased estimates of the variance of the response data explained by the MEM eigenfunctions. The results are also presented in the form of a graph (Fig. S4.2).

Table S4.1. Adjusted R-squares resulting from analysis against MEM eigenfunctions of the fitted and residual response distances, obtained by regression on three transformations of the geographic distances: raw geographic distances  $D(XY)$ , square-rooted distances  $\sqrt{D(XY)}$ , and  $\log_e$ -transformed distances  $\ln(D(XY))$ .

	Fitted.D_adj.R2	Residual.D_adj.R2
D(XY)	0.08432	0.45886
$\sqrt{D(XY)}$	0.14596	0.51057
$\ln(D(XY))$	0.26962	0.60410

Firstly and foremost, the graph shows that the fitted distances account for a rather small fraction (blue symbols) of the spatial structure of the generated data that can be modelled by MEM eigenfunctions (black horizontal line). A surprisingly high amount of spatial correlation can be modelled by the same MEM eigenfunctions from the residual distances. Regression on any of the three types of geographic distance matrices has not controlled for the spatial structure in the response data since it has left a lot of that structure in the residual distances. Secondly, the graph shows that the square-root and log transformations of the geographic distances, before regression (or Mantel test), offer a slightly better performance in terms of capturing the spatial

autocorrelation of the response data than the original representation of the geographic relationships as a simple distance matrix computed from the geographic coordinates,  $D(XY)$ , although we observe the surprising consequence that there is also more spatial autocorrelation left in the residuals with the transformed distances. This seems to be due to a distortion of the distances when regressing  $D(Y)$  on  $\sqrt{D(XY)}$  or  $\ln(D(XY))$ , which causes the appearance of spurious broad-scaled spatial structures in both the fitted values and the residuals. These structures can be identified as broad-scaled because they can be modelled by the first few largest-scaled MEM eigenfunctions. These eigenfunctions model the broad-scaled structures generated in the fitted values and residuals of the regressions on  $\sqrt{D(XY)}$  and  $\ln(D(XY))$  more strongly (meaning: higher  $R^2$ ) than in the fitted values and residuals of the regression on  $D(XY)$ . Thus, regression on distance matrices is no better using  $\sqrt{D(XY)}$  or  $\ln(D(XY))$  than using  $D(XY)$ . A more detailed investigation of this phenomenon would be in order.

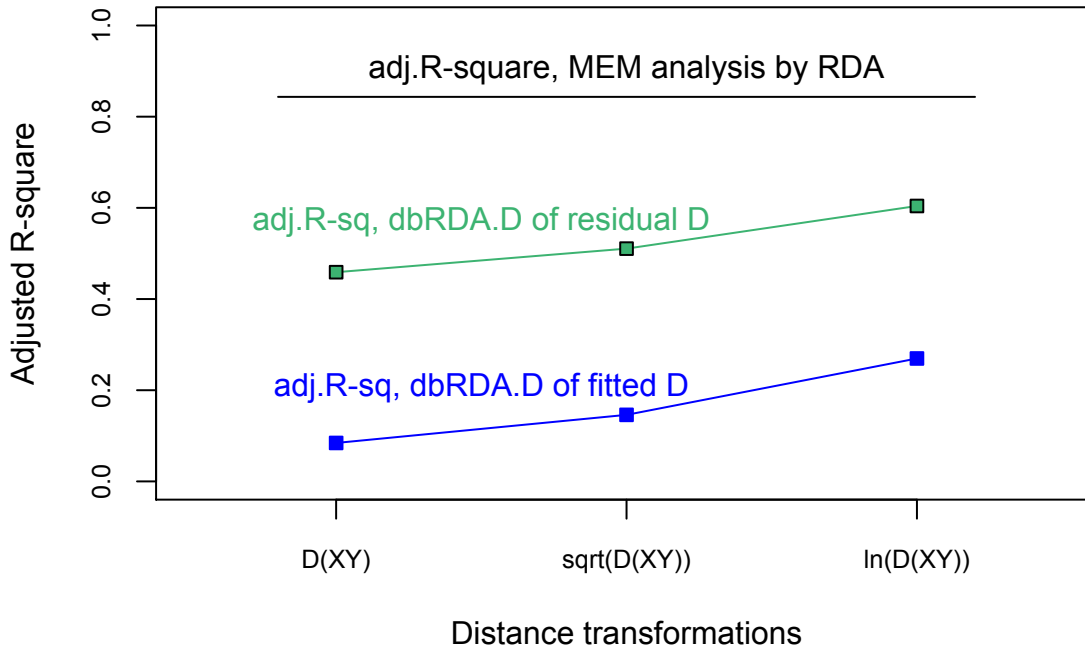


Fig. S4.2. Adjusted R-squares resulting from analysis of the fitted (blue) and residual response distances (green), obtained by regression on three transformations of the geographic distances (abscissa), against MEM eigenfunctions: raw geographic distances  $D(XY)$ , square-rooted distances  $\sqrt{D(XY)}$ , and  $\log_e$ -transformed distances  $\ln(D(XY))$ . The adjusted R-square of the original regression analysis of the generated data against the MEM eigenfunctions is also presented (black horizontal line in the graph); this is the amount of explained variance that can be extracted from the response data by MEM analysis.

In any case, using three representations of the geographic relationships by distance matrices that are commonly used to represent spatial relationships in Mantel tests, the example has shown that the residual distances *did* contain significant spatial autocorrelation (SA). Hence, regression of the response distances on the geographic distance matrices did not “remove” or “control for”

spatial correlation in the response data, as is often assumed by users of Mantel tests or regression on distance matrices.

Readers are invited to generate new response data with the function `gen.SA.data()` and analyse them at leisure.

## References

- Borcard, D. & Legendre, P. (2002) All-scale spatial analysis of ecological data by means of principal coordinates of neighbour matrices. *Ecological Modelling*, **153**, 51–68.
- Borcard, D., Legendre, P., Avois-Jacquet, C. & Tuomisto, H. (2004) Dissecting the spatial structure of ecological data at multiple scales. *Ecology*, **85**, 1826–1832.
- Dray, S., P. Legendre, P. & Peres-Neto, P. R. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modelling*, **196**, 483–493.
- Legendre, P., Borcard, D. & Peres-Neto, P. R. (2005) Analyzing beta diversity: partitioning the spatial variation of community composition data. *Ecological Monographs*, **75**, 435–450.
- Legendre, P. & Fortin, M.-J. (2010) Comparison of the Mantel test and alternative approaches for detecting complex multivariate relationships in the spatial analysis of genetic data. *Molecular Ecology Resources*, **10**, 831–844.
- Legendre, P. & Legendre, L. (2012) *Numerical ecology. 3rd English edition*. Elsevier Science BV, Amsterdam.
- McArdle, B. H. & Anderson, M. J. (2001) Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology*, **82**, 290–297.

# Generation of random SA data on a square grid

## Description

This function generates random spatially autocorrelated (SA) data on a square grid with  $nx$  points in the X and Y directions and analyses it with dbMEM eigenfunctions and Mantel tests.

## Usage

```
gen.SA.data(nx=20, surf=NULL, range=5, var=1, nperm1=0, nperm2=999)
```

## Arguments

<code>nx</code>	Number of rows and columns of the regular grid; $n = nx^2$ .
<code>surf</code>	Vector with values for each point of the response surface. The vector may be a <code>surf</code> vector produced during a previous run of the function, or contain real univariate data about a surface forming a regular square grid. Default: <code>surf=NULL</code> .
<code>range</code>	Range parameter of the variogram for generation of SA on the surface.
<code>var</code>	Total variance of the simulated data.
<code>nperm1</code>	Number of permutations for dbMEM analysis by regression. Default: <code>nperm1=0</code> .
<code>nperm2</code>	Number of permutations for the Mantel tests. Default: <code>nperm2=999</code> .

## Details

Required packages:

- Load the {RandomFields} and {vegan} available from CRAN.
- Load package {PCNM}. The source code (for Linux) and compiled Windows file are available from [https://r-forge.r-project.org/R/?group\\_id=195](https://r-forge.r-project.org/R/?group_id=195). A Mac OSX file compiled for R 3.0.x is available on <http://adn.biol.umontreal.ca/~numericaledcology/Rcode/>.

If "surf" is NULL, a surface is simulated by Gaussian random fields controlled by a spherical variogram:

If "surf" is not NULL and a "surf" vector of the correct size is provided ( $length=nx^2$ ), containing values for each point of the response surface, the dbMEM and Mantel analyses will be carried out on that surface.



## Value

Function `gen.SA.data` returns a list containing the following results:

<code>R2.dbMEM</code>	R-square and adjusted R-square of dbMEM analysis by regression.
<code>p.dbMEM</code>	p-values (parametric and permutational) of dbMEM analysis by regression.
<code>R2.Mantel</code>	Mantel correlations squared with <code>geo.D</code> , <code>sqrt(geo.D)</code> and <code>log(geo.D)</code> .
<code>p.Mantel</code>	Mantel test p-values with <code>geo.D</code> , <code>sqrt(geo.D)</code> and <code>log(geo.D)</code> .
<code>Surface</code>	The generated surface presented as a matrix.
<code>surf</code>	The generated data presented as a vector.
<code>grid.coord</code>	A matrix with grid coordinates, from which <code>D.geo</code> can be recomputed.
<code>dbMEM</code>	The matrix of dbMEM spatial eigenvectors.

## References

Legendre, P. & Legendre, L. (2012) *Numerical Ecology, 3rd English edition*. Elsevier Science BV, Amsterdam.

Schlather, M., Malinowski, A., Oesting, M., Boecker, D., Strokorb, K., Engelke, S., Martini, J., Menck, P., Gross, S., Burmeister, K., Manitz, J., Singleton, R., Pfaff, B. and R Core Team (2014). *RandomFields: Simulation and Analysis of Random Fields*. R package version 3.0.10. <http://CRAN.R-project.org/package=RandomFields>

## Author

Pierre Legendre, Département de sciences biologiques, Université de Montréal  
License: GPL (>=2)

## Example

```
# Generate a new random 20x20 spatially autocorrelated (SA) surface with variogram range = 5.
```

```
res = gen.SA.data(nx=20, range=5, var=1, nperm1=999, nperm2=999)
```

```
# Reanalyse the surface generated during the previous run. A new surface is not generated if
parameter surf is not NULL. The surf vector provided must be of the correct length, nx^2.
```

```
res2 = gen.SA.data(nx=20, surf=res$surf, range=5, var=1, nperm1=999, nperm2=999)
```

```
# The vector may be a surf vector produced during a previous run of the function, as in this
example, or contain univariate data on some other surface forming a regular square grid.
```

---

```
gen.SA.data <- function(nx=20, surf=NULL, range=5, var=1, nperm1=0, nperm2=999)
{
# Load packages
  require(RandomFields)
  require(vegan)
  require(PCNM)
  cat("Function RFsimulate() of RandomFields 3.0.10 is used with R 3.0\n")
# Package Random Fields options
  RFOptions(spConform=FALSE)
  RFOptions(seed=NULL)
#
a <- system.time({          # How much time for the calculations?

  coord <- 1:nx
  n <- nx^2
  grid.coord <- expand.grid(coord, coord)
  colnames(grid.coord) <- c("Easting", "Northing")
#
# If no surface is provided, one is generated here
  if(is.null(surf)) {
    surf <- vector(mode="numeric", length=n)
    ## Simulate autocorrelated surface with spherical variogram model
    model = RMSpheric(var=var, scale=range)
    Surface <- RFsimulate(model = model, x=coord, y=coord, grid=TRUE)
    ### If a nugget is sought, argument err.model in RFsimulate
    ### can be used: err.model=RMnugget(var=var)
    #
    for(k in 1:n) surf[k] <- Surface[grid.coord[k,2], grid.coord[k,1]]
  } else {
# else, the "surf" data provided are used for the analysis
    if(length(surf) != n) stop("The 'surf' vector provided is not of length
nx^2")
    Surface <- matrix(surf,nx,nx,byrow=TRUE)
  }
  Y.D <- dist(surf)
#
# Construct dbMEM eigenfunctions. Keep the eigenfunctions modelling positive SA
  geo.D <- dist(grid.coord)
  dbMEM.grid <- PCNM(geo.D,thresh=1,dbMEM=TRUE)
  dbMEM <- dbMEM.grid$vectors
#
# dbMEM analysis by regression. No selection of the eigenfunctions
  lm.res <- lm(surf ~ .,data=as.data.frame(dbMEM))
  R2.dbMEM <- c(summary(lm.res)$r.squared, summary(lm.res)$adj.r.squared)
  F.vec <- summary(lm.res)$fstatistic
#
# Compute the regression parametric p.value
  p.dbMEM.temp <- pf(F.vec[1], F.vec[2], F.vec[3], lower.tail=FALSE)[[1]]
#
  if(nperm1==0) {
    # Output only the parametric regression test p.value
    p.dbMEM <- c(p.dbMEM.temp, NA)
  } else {
    # Output the parametric and permutation test p.values
    p.dbMEM <- c(p.dbMEM.temp,
R2.test.perm(surf,dbMEM,nperm=nperm1,dbMEM=TRUE)$P)
  }
# Mantel test using vegan's mantel() function
  res.Mantel1 <- mantel(Y.D, geo.D, permutations=nperm2)
  res.Mantel2 <- mantel(Y.D, sqrt(geo.D), permutations=nperm2)
```

```
res.Mantel3 <- mantel(Y.D, log(geo.D), permutations=nperm2)
R2.Mantel <-
  c(res.Mantel1$statistic^2, res.Mantel2$statistic^2,
res.Mantel3$statistic^2)
p.Mantel <- c(res.Mantel1$signif, res.Mantel2$signif, res.Mantel3$signif)
})
a[3] <- sprintf("%2f",a[3])
cat("Calculation time =",a[3]," sec",'\\n')
#
list(R2.dbMEM=R2.dbMEM, p.dbMEM=p.dbMEM, R2.Mantel=R2.Mantel, p.Mantel=p.Mantel,
Surface=Surface, surf=surf, grid.coord=grid.coord, dbMEM=as.data.frame(dbMEM))
}

R2.test.perm <- function(Y, X, nperm=999, dbMEM=FALSE)
#
# Permutation test for R2 statistic in regression or RDA.
#
# Parameters of the function --
# Y : Matrix or vector of response data.
# X : Matrix of explanatory data, e.g. a file of dbMEM.
# nperm : Number of permutations for the test.
# dbMEM=FALSE: Normal computation of statistics for regression or RDA:
#               Centre X, QR decomposition of X, compute adjusted R-square.
# dbMEM=TRUE : Do not centre X since the matrix of dbMEM is centred.
#               Do not compute the adjusted R-square.
# License: GPL-2
# Author:: Pierre Legendre, June 2014
{
  Y.c <- scale(Y,center=TRUE,scale=FALSE)          # Centre Y
  SS.Y <- sum(Y.c^2)
  X <- as.matrix(X)
  n <- nrow(X)
  if(!dbMEM) X <- scale(X,center=TRUE,scale=FALSE) # Centre X if not dbMEM
  Q <- qr(X)                                       # QR decomposition of X
  Yfit.X <- qr.fitted(Q, Y.c)                     # Compute fitted values
# Compute statistics
  Rsquare <- sum(Yfit.X^2)/SS.Y
  m <- Q$rank
  residualDF <- n-m-1
  F <- (Rsquare*residualDF)/((1-Rsquare)*m)
  if(dbMEM) { adjRsqr <- NA
    } else {
    totalDF <- n-1
    adjRsqr <- 1-((1-Rsquare)*totalDF/residualDF) }
# Permutation test of F
  if(nperm > 0) {
    nPGE <- 1
    for(i in 1:nperm) {
      YfitPerm <- qr.fitted(Q, sample(Y.c))
      RsquarePerm <- sum(YfitPerm^2)/SS.Y
      FPerm <- (RsquarePerm*residualDF)/((1-RsquarePerm)*m)
      if(FPerm >= F) nPGE <- nPGE+1
    }
    P <- nPGE/(nperm+1)
  } else { P <- NA }
#
if(dbMEM) { out <- list(P=P)
  } else {
  out <- list(Rsquare=Rsquare, F=F, P=P, adjRsqr=adjRsqr, nperm=nperm, m=m,
  residualDF=residualDF) }
}
```

## dbRDA $F$ -test for response data in dissimilarity matrix form

### Description

Compute the dbRDA  $F$ -test of significance between response data represented by a Euclidean or non-Euclidean dissimilarity matrix and a matrix of explanatory variables, using the method of McArdle and Anderson (2001).

### Usage

```
dbRDA.D(D, X, nperm=999, option=3, compute.eig=FALSE, coord=FALSE,  
        rda.coord=2, pos.RDA.val=FALSE)
```

### Arguments

- |             |  |
|-------------|--|
| D           | Dissimilarity matrix (class <code>matrix</code> or <code>dist</code> ) representing the response data. D may be Euclidean or non-Euclidean.  |
| X           | Rectangular matrix of explanatory variables for the RDA (class <code>data.frame</code> or <code>matrix</code> ). Factors must be recoded as dummy variables or Helmert contrasts.    |
| option      | =1 : Original McArdle-Anderson (2001) equation 4. Slow, not recommended.<br>=2 : McArdle-Anderson equation, simplified.<br>=3 : Least-squares solution after orthogonalization of X. |
| compute.eig | =TRUE : the eigenvalues and eigenvectors of D are computed. Do <i>not</i> use with very large matrices (slow).   |
| coord       | =TRUE : compute the principal coordinates corresponding to the positive eigenvalues of D. This option requires <code>compute.eig=TRUE</code> .                                       |
| rda.coord   | Number of RDA ordination coordinates to compute, for example 2 (default value).  |
| pos.RDA.val | =TRUE : store only positive RDA eigenvalues in the output list.<br>=FALSE : store all RDA eigenvalues in the output list.  |

### Details

Compute the dbRDA  $F$ -test of significance. The response is represented by a Euclidean or non-Euclidean dissimilarity matrix; X is a matrix of explanatory variables, as in regular RDA.

The  $F$ -statistic is obtained without prior computation of the eigenvalues and eigenvectors of the dissimilarity matrix, hence no correction has to be made to eliminate the negative eigenvalues. Three computation methods are available, all derived from McArdle and Anderson (2001).

The eigenvalues and eigenvectors of D are computed if `compute.eig=TRUE`. If `coord=TRUE`, the principal coordinates corresponding to the positive eigenvalues of D are computed.

The function may fail to produce a meaningful RDA test of significance and ordination axes if  $D$  is extremely non-Euclidean. This is the case with some forms of genomic distances.

Computation options:

option=1 — The original  $F$ -statistic of McArdle and Anderson (2001), eq. 4:

$$F = \text{SSYhat} / \text{sum}(\text{diag}(\mathbf{I} - \mathbf{H} \%*\% \mathbf{G} \%*\% \mathbf{I} - \mathbf{H}))$$

Degrees of freedom are added to this equation when writing the output list.

option=2 — Simplified equation:

$$F = \text{SSYhat} / (\text{SSY} - \text{SSYhat})$$

option=3 — Orthogonalize matrix  $X$  by PCA before computing projector  $H$ . No inversion.

Compute  $\text{SSYhat}$ , then  $F = \text{SSYhat} / (\text{SSY} - \text{SSYhat})$

Options 2 and 3 are equivalent; they require half the computing time of option 1.

## Value

Function `dbRDA.D` returns a list containing the following results:

<code>F</code>	$F$ -statistic.
<code>Rsquare</code>	R-square and adjusted R-square statistics.
<code>P.perm</code>	Permutational p-value of RDA R-square (test based on $F$ ).
<code>SS.total</code>	Trace of Gower-centred matrix $G$ . The trace is equal to the total sum of squares of $Y$ and to the sum of the eigenvalues of $D$ .
<code>PCoA.values</code>	Eigenvalues (if they are computed, i.e. if <code>compute.eig=TRUE</code> ).
<code>PCoA.vectors</code>	Principal coordinates for the positive eigenvalues of $D$ .
<code>RDA.values</code>	RDA eigenvalues.
<code>RDA.rel.values</code>	RDA relative eigenvalues.
<code>RDA.cum.values</code>	RDA cumulative relative eigenvalues.
<code>RDA.coord</code>	Ordination coordinates of objects on selected RDA axes.

## References

- Legendre, P. (2014) Interpreting the replacement and richness difference components of beta diversity. *Global Ecology and Biogeography*, **23**, 1324–1334.
- Legendre, P. & Legendre, L. (2012) *Numerical Ecology*, 3rd English edition. Elsevier Science BV, Amsterdam.
- McArdle, B.H. & Anderson, M.J. (2001) Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology*, **82**, 290–297.

## Author

Pierre Legendre, Département de sciences biologiques, Université de Montréal

License: GPL ( $\geq 2$ )

## Example

# Load function dbRDA.D()

# 1. Analysis of mite data with the percentage difference (*alias* Bray-Curtis) dissimilarity

```
require(vegan)
data(mite)
data(mite.env)                # The first 2 environmental variables are quantitative
sel = c(14,24,31,41,49,64)    # Select 6 sites for the example
mite.BC = vegdist(mite[sel,], "bray") # The D matrix will produce one negative eigenvalue
res = dbRDA.D(mite.BC, mite.env[sel,1:2], nperm=999, compute.eig=TRUE)
res$Rsquare
res$P.perm
# Plot the ordination on PCoA axes 1 and 2
plot(res$RDA.coord)
text(res$RDA.coord, labels=rownames(mite.env[sel,]), pos=3)
```

# 2. Compare RDA to dbRDA.D results using Euclidean distance

```
mite.hel = decostand(mite, "hellinger")
mite.hel.D = dist(mite.hel)
# RDA solution
rda.out = rda(mite.hel ~ SubsDens+WatrCont, data=mite.env)
RsquareAdj(rda.out)
anova(rda.out, step=1000, perm.max=1000)
# dbRDA.D solution
dbRDA.out = dbRDA.D(mite.hel.D, mite.env[,1:2], nperm=999, compute.eig=TRUE)
dbRDA.out$F
dbRDA.out$Rsquare
dbRDA.out$P.perm
```

---

```
dbRDA.D <- function(D, X, nperm=999, option=3, compute.eig=FALSE, coord=FALSE,
rda.coord=2, positive.RDA.values=FALSE)
{
  D <- as.matrix(D)
  X <- as.matrix(X)
  n <- nrow(D)
  epsilon <- .Machine$double.eps
#
# Gower centring, matrix formula. Legendre & Legendre (2012), equation 9.42
  One <- matrix(1,n,n)
  mat <- diag(n) - One/n
  G <- -0.5 * mat %*% (D^2) %*% mat
  SSY <- sum(diag(G))
  # LCBD <- diag(G)
#
# Principal coordinate analysis after eigenvalue decomposition of D
  if(compute.eig) {
    eig <- eigen(G, symmetric=TRUE)
    values <- eig$values      # All eigenvalues
    vectors <- eig$vectors    # All eigenvectors, scaled to lengths 1
    if(coord) {
      select <- which(values > epsilon)
      princ.coord <- vectors[,select] %*% diag(sqrt(values[select]))
    } else { princ.coord <- NA }
  } else {
    values <- princ.coord <- NA
  }
#
# Compute projector matrix H ("hat" matrix in the statistical literature)
  X.c <- scale(X, center=TRUE, scale=FALSE) # Centre matrix X
  m <- qr(X.c, tol=1e-6)$rank                # m = rank of X.c
  cat("Rank of X centred =",m,"\n")
  if(m==1) {
    H <- (X.c[,1] %*% t(X.c[,1]))/((t(X.c[,1]) %*% X.c[,1])[1,1])
  } else {
    if(option<3) {
      # if(det(t(X.c)%*%X.c)<epsilon) stop ('Collinearity detected in X')
      if(m < ncol(X.c)) stop ('Collinearity detected in X')
      H <- X.c %*% solve(t(X.c) %*% X.c) %*% t(X.c)
      #
      # option=3: compute projector H from orthogonalized X; no inversion
    } else {
      X.eig <- eigen(cov(X.c))
      k <- length(which(X.eig$values > epsilon))
      X.ortho <- X.c %*% X.eig$vectors[,1:k] # F matrix of PCA
      XprX <- t(X.ortho) %*% X.ortho
      H <- X.ortho %*% diag(diag(XprX)^(-1)) %*% t(X.ortho)
    }
  }
#
# Compute the F statistic: McArdle & Anderson (2001), equation 4 modified
  HGH <- H %*% G %*% H
  SSYhat <- sum(diag(HGH))
#
  if(option==1) {
    I.minus.H <- diag(n) - H
    den1 <- sum(diag(I.minus.H %*% G %*% I.minus.H))
    F <- SSYhat/den1      # F statistic without the degrees of freedom
    Rsquare <- F/(F+1)
  } else {
```



```

      F <- SSYhat/(SSY-SSYhat) # F statistic without the degrees of freedom
      Rsquare <- SSYhat/SSY    # or equivalent: Rsquare <- F/(F+1)
    }
    RsqAdj <- 1-((1-Rsquare)*(n-1)/(n-1-m))
#
# Permutation test of F
  if(nperm > 0) {
    nGE=1
    for(i in 1:nperm) {
      order <- sample(n)
      Gperm <- G[order, order]
      H.Gperm.H <- H %%% Gperm %%% H
      SSYhat.perm <- sum(diag(H.Gperm.H))
      #
      if(option==1) {
        den <- sum(diag(I.minus.H %%% Gperm %%% I.minus.H))
        F.perm <- SSYhat.perm/den
      } else {
        F.perm <- SSYhat.perm/(SSY-SSYhat.perm)
      }
      if(F.perm >= F) nGE=nGE+1
    }
    P.perm <- nGE/(nperm+1)
  } else { P.perm <- NA }
#
# Compute RDA ordination coordinates
  if(rda.coord > 0) {
    HGH.eig <- eigen(HGH, symmetric=TRUE)
    # kk <- length(which(HGH.eig$values > epsilon))
    RDA.values <- HGH.eig$values
    rel.eig <- RDA.values/SSY
    cum.eig <- cumsum(rel.eig)
    kk <- length(which(rel.eig > epsilon))
    if(positive.RDA.values) {
      RDA.values <- RDA.values[1:kk]
      rel.eig <- rel.eig[1:kk]
      cum.eig <- cum.eig[1:kk]
    }
    k <- min(rda.coord, kk)
    if(k >= 2) {
      RDA.coord <- sweep(HGH.eig$vectors[,1:k], 2, sqrt(RDA.values[1:k]), FUN="*")
    } else {
      RDA.coord <- NA
      cat("k =", k, " -- Fewer than two RDA eigenvalues > 0\n")
    }
  } else { RDA.values <- rel.eig <- cum.eig <- RDA.coord <- NA }
#
list(F=F*(n-m-1)/m, Rsquare=c(Rsquare, RsqAdj), P.perm=P.perm, SS.total=SSY,
PCoA.values=values, PCoA.vectors=princ.coord, RDA.values=RDA.values/(n-1),
RDA.rel.values=rel.eig, RDA.cum.values=cum.eig, RDA.coord=RDA.coord)
}

```

## Appendix S5

### SOFTWARE USED IN THE SIMULATIONS

This Appendix presents the software used in the simulations reported in section “Simulations involving spatially autocorrelated data: comparison of Mantel test and dbMEM analysis” of the paper. A separate R function was written for each series of simulations. This function was called by a set of R commands that produced output files. These files of R commands were run in batch mode.

Files shows in the following pages:

*Series 1 simulations involving all pairwise geographic distances* (see paper, Fig. 2)

- Simulation function: file LFB.simul1.R
- Running the simulation function: file run.LFB.simul1.batch.txt

*Series 2 simulations involving truncated geographic distance matrices* (see paper, Fig. 3)

- Simulation function: file LFB.simul2.R
- Running the simulation function: files run.LFB.simul2.batch1.txt, run.LFB.simul2.batch2.txt, run.LFB.simul2.batch3.txt, run.LFB.simul2.batch4.txt

*Series 3 simulations involving Delaunay triangulations* (see Appendix S3, Fig. S3.1)

- Simulation function: file LFB.simul3.R
- Running the simulation function: files run.LFB.simul3.batch1.txt, run.LFB.simul3.batch2.txt, run.LFB.simul3.batch3.txt, run.LFB.simul3.batch4.txt

The simulation functions call upon packages RandomFields, vegan and spatstat available on CRAN, and package PCNM available on [http://r-forge.r-project.org/R/?group\\_id=195](http://r-forge.r-project.org/R/?group_id=195) for Linux and Windows versions, and on <http://adn.biol.umontreal.ca/~numericaledgeology/FonctionsR/> for a Mac OSX compiled version.

```
LFB.simul1 <- function(nsim=100, spacing=3, range=5, mean=0.0, nugget=0.0, var=1, nperm1=99,
nperm2=99, R.option=2)
#
# Task: Simulation function for the Legendre, Fortin & Borcard paper.
# Map sizes are chosen to harbour 100 sampling units (n=100) with spacing of
# {1,2,3,4,5} with a buffer zone of 5 pixels all around the sampled area.
#
# Parameters of the function --
#
# nsim : number of surfaces to be simulated.
# spacing = {1,2,3,4,5} for size of simulation surface = {20,29,38,47,56}.
# range : range of the variogram for generation of SA on the surface.
# mean : mean of the simulated data
# nugget: nugget parameter
# var : total variance of the simulated data; var = nugget + sill.
# nperm1 : number of permutations for dbMEM analysis.
# nperm2 : number of permutations for Mantel analysis.
#
# Simulation of Gaussian random fields controlled by a spherical variogram:
# R.option=2 : With R 2.15, use function GaussRF() of RandomFields 2.0.66
# R.option=3 : With R 3.0, use function RFsimulate() of RandomFields 3.0.10
#
# Permutational test in dbMEM analysis: see function R2.test.perm().
#
# Value (output list) --
#
# rej.dbMEM : rejection rates, dbMEM analysis, parametric and permutation tests.
# rej.Mantel : rejection rates, Mantel analysis, permutation test.
# R.dbMEM : R-square & adj. R-square of indiv. simulations dbMEM anal. (matrix).
# p.dbMEM : p-values (param. & perm.) of individual simul. dbMEM anal. (matrix).
# r.Mantel : Mantel correlation of individual simulations with geoD and sqrt(geo.D).
# p.Mantel : p-values of individual simulations, Mantel with geoD and sqrt(geo.D).
# param : vector listing the run parameters: {nsim,spacing,range,nperm1,nperm2}.
# geo.D : geographic distance matrix for the grid sample, shown only if nsim=1.
# Y.D : response distance matrix for the grid sample, shown only when nsim=1.
# Surface : matrix with the whole response surface, shown only when nsim=1.
#           A map can be plotted using image(x$Surface)
#
# References --
#
# Schlather, M., P. Menck, R. Singleton, B. Pfaff and R Core team (2013).
#   RandomFields: Simulation and analysis of random fields.
#   R package version 2.0.66. http://CRAN.R-project.org/package=RandomFields
#
# Schlather, M., A. Malinowski, M. Oesting, D. Boecker, K. Strokorb, S. Engelke,
#   J. Martini, P. Menck, S. Gross, K. Burmeister, J. Manitz, R. Singleton,
#   B. Pfaff and R Core Team (2014). RandomFields: Simulation and Analysis
#   of Random Fields. R package version 3.0.10.
#   http://CRAN.R-project.org/package=RandomFields
#
# License: GPL-2
# Authors: Pierre Legendre and D. Borcard, June 2014
{
require(RandomFields)
require(vegan)
require(PCNM)
param <-c(nsim=nsim, spacing=spacing, range=range, nperm1=nperm1, nperm2=nperm2)
#
if(R.option==2) {
  cat("Function GaussRF() of RandomFields 2.0.66 is used with R 2.15\n")
} else if(R.option==3) {
  cat("Function RFsimulate() of RandomFields 3.0.10 is used with R 3.0\n")
  Rfoptions(spConform=FALSE)
  Rfoptions(seed=NULL)
} else { stop("Error in R.option parameter") }
```

```
#
# Set up the simulation surface and the sampling grid parameters
size <- 9*spacing+11      # size = {20,29,38,47,56} depending on "spacing"
coo <- 1:size            # Coordinates of points on simulation surface in each direction
#
# Coordinates of the sample of n=100 points on the surface
grid.coord <- expand.grid(seq(6,(size-5),spacing), seq(6,(size-5),spacing))
n <- nrow(grid.coord)
geo.D <- dist(grid.coord)
#
# Construct dbMEM eigenfunctions
dbMEM.grid <- PCNM(geo.D,thresh=spacing,dbMEM=TRUE)
dbMEM <- as.data.frame(dbMEM.grid$vectors)
Q <- qr(as.matrix(dbMEM))
#
# Prepare matrix and vectors to receive the simulation results
surf <- vector(mode="numeric", length=n)
R.dbMEM <- matrix(NA,nsim,2)
colnames(R.dbMEM) <- c("R.square","adj.R.square")
p.dbMEM <- matrix(NA,nsim,2)
colnames(p.dbMEM) <- c("Parametric.P","Permutational.P")
r.Mantel <- matrix(NA,nsim,2)
p.Mantel <- matrix(NA,nsim,2)
colnames(r.Mantel) <- c("Mantel(Y.D,geo.D)","Mantel(Y.D,sqrt(geo.D))")
colnames(p.Mantel) <- c("Mantel(Y.D,geo.D)","Mantel(Y.D,sqrt(geo.D))")
###
# Main simulation loop
a <- system.time({      # How much time for the simulations?
#
for(kk in 1:nsim) {
  if(range==0) {
    surf <- rnorm(n)
  } else {
    ## Simulate autocorrelated surface with spherical variogram model
    if(R.option==2) {
      Surface <- GaussRF(coo, coo, model="spherical", grid=TRUE,
        param=c(mean=mean, var=var, nugget=nugget, scale=range))
    }
    if(R.option==3) {
      model = RMspheric(var=var, scale=range)
      Surface <- RFsimulate(model = model, x=coo, y=coo, grid=TRUE)
      ### DB: if a nugget is sought, argument err.model in RFsimulate
      ### can be used: err.model=RMnugget(var=var)
    }
    #
    for(k in 1:n) surf[k] <- Surface[grid.coord[k,2], grid.coord[k,1]]
  }
  Y.D <- dist(surf)
  #
  # dbMEM analysis by regression. No selection of the eigenfunctions
  lm.res <- lm(surf ~ .,data=dbMEM)
  R.dbMEM[kk,] <- c(summary(lm.res)$r.squared, summary(lm.res)$adj.r.squared)
  F.vec <- summary(lm.res)$fstatistic
  # Parametric test results (for the time being)
  p.dbMEM[kk,1] <- pf(F.vec[1], F.vec[2], F.vec[3], lower.tail=FALSE)
  if(nperm1==0) { p.dbMEM[kk,2] <- NA } else {
    p.dbMEM[kk,2] <- R2.test.perm(surf,dbMEM,nperm=nperm1,Q=Q, dbMEM=TRUE)$P }
  # Mantel test using vegan's mantel() function
  res.Mantel1 <- mantel(Y.D, geo.D, permutations=nperm2)
  res.Mantel2 <- mantel(Y.D, sqrt(geo.D), permutations=nperm2)
  r.Mantel[kk,1] <- res.Mantel1$statistic
  r.Mantel[kk,2] <- res.Mantel2$statistic
  p.Mantel[kk,1] <- res.Mantel1$signif
  p.Mantel[kk,2] <- res.Mantel2$signif
}
}
rej.dbMEM <-
```

```

      c(length(which(p.dbMEM[,1]<=0.05)),length(which(p.dbMEM[,2]<=0.05)))/nsim
rej.Mantel <- c(length(which(p.Mantel[,1]<=0.05)),length(which(p.Mantel[,2]<=0.05)))/nsim
#
})
a[3] <- sprintf("%2f",a[3])
cat("Simulation time =",a[3]," sec",'\\n')
#
# Save Surface, geo.D and Y.D only if a single simulation has been produced
if(nsim==1) {Surface<-Surface; geo.D=geo.D; Y.D<-Y.D}
  else {Surface<-NULL; geo.D=NULL; Y.D<-NULL}
#
out <- list(rej.dbMEM=rej.dbMEM, rej.Mantel=rej.Mantel, R.dbMEM=R.dbMEM, p.dbMEM=p.dbMEM,
r.Mantel=r.Mantel, p.Mantel=p.Mantel, param=param, geo.D=geo.D, Y.D=Y.D, Surface=Surface)
}

R2.test.perm <- function(Y, X, nperm=999, Q=NULL, dbMEM=FALSE)
# Permutation test for R2 statistic in regression or RDA
# Some operations on X=dbMEM are transferred to the main function LFB.simul1()
# in the case of dbMEM analysis.
#
# Parameters of the function --
# Y : Matrix or vector of response data.
# X : Matrix of explanatory data, e.g. a file of dbMEM.
# nperm : Number of permutations for the test.
# dbMEM=FALSE: Normal computation of statistics for regression or RDA:
#               Centre X, QR decomposition of X, compute adjusted R-square.
# dbMEM=TRUE : Do not centre X since the matrix of dbMEM is centred.
#               Do not compute the adjusted R-square.
# License: GPL-2
# Author:: Pierre Legendre, June 2014
{
Y.c <- scale(Y,center=TRUE,scale=FALSE)    # Centre Y
SS.Y <- sum(Y.c^2)
X <- as.matrix(X)
n <- nrow(X)
if(!dbMEM) {
  X <- scale(X,center=TRUE,scale=FALSE)    # Centre X
  Q <- qr(X)                               # QR decomposition of X
}
Yfit.X <- qr.fitted(Q, Y.c)
# Compute statistics
Rsquare <- sum(Yfit.X^2)/SS.Y
m <- Q$rank
residualDF <- n-m-1
F <- (Rsquare*residualDF)/((1-Rsquare)*m)
if(dbMEM) { adjRsqr <- NA } else {
  totalDF <- n-1
  adjRsqr <- 1-((1-Rsquare)*totalDF/residualDF)
}
# Permutation test of F
if(nperm > 0) {
  nPGE <- 1
  for(i in 1:nperm) {
    YfitPerm <- qr.fitted(Q, sample(Y.c))
    RsquarePerm <- sum(YfitPerm^2)/SS.Y
    FPerm <- (RsquarePerm*residualDF)/((1-RsquarePerm)*m)
    if(FPerm >= F) nPGE <- nPGE+1
  }
  P <- nPGE/(nperm+1)
} else { P <- NA }
#
if(dbMEM) { out <- list(P=P) }
else {out <- list(Rsquare=Rsquare, F=F, P=P, adjRsqr=adjRsqr, nperm=nperm, m=m,
residualDF=residualDF) }
}

```

File **run.LFB.simul1.batch.txt** — Batch run for series 1 simulations

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul1.R"  
source("LFB.simul1.R")
```

```
res1.5.0 = LFB.simul1(nsim=1000, spacing=5, range=0, nperm1=999, nperm2=999)  
#  
res1.5.5 = LFB.simul1(nsim=1000, spacing=5, range=5, nperm1=999, nperm2=999)  
#  
res1.5.10 = LFB.simul1(nsim=1000, spacing=5, range=10, nperm1=999, nperm2=999)  
#  
res1.5.15 = LFB.simul1(nsim=1000, spacing=5, range=15, nperm1=999, nperm2=999)  
#  
res1.5.20 = LFB.simul1(nsim=1000, spacing=5, range=20, nperm1=999, nperm2=999)  
#  
res1.5.25 = LFB.simul1(nsim=1000, spacing=5, range=25, nperm1=999, nperm2=999)  
#  
res1.5.30 = LFB.simul1(nsim=1000, spacing=5, range=30, nperm1=999, nperm2=999)  
#  
res1.5.35 = LFB.simul1(nsim=1000, spacing=5, range=35, nperm1=999, nperm2=999)  
#  
res1.5.40 = LFB.simul1(nsim=1000, spacing=5, range=40, nperm1=999, nperm2=999)  
#  
save(res1.5.0,res1.5.5,res1.5.10,res1.5.15,res1.5.20,res1.5.25,res1.5.30,res1.5.35,res1.5.40,  
file="simul1,spacing=5.9_runs_sqrt.RData")
```

---

```
LFB.simul2 <- function(nsim=100, spacing=5, range=10, thresh=20, maxD=NULL, mean=0.0,
nugget=0.0, var=1, nperm1=99, nperm2=99, R.option=2)
#
# Task: Simulation function for the Legendre, Fortin & Borcard paper.
# Map sizes are chosen to harbour 100 sampling units (n=100) with spacing of
# {1,2,3,4,5} with a buffer zone of 5 pixels all around the sampled area.
#
# In the LFB.simul2 simulations, autocorrelated surfaces are generated with different
# range values, as in the LFB.simul1 function. However, instead of considering the whole
# matrix of geographic distances among points, the distances larger than the truncation
# distance ("thresh") are changed to the largest distance in the data set (maxD). A Mantel
# test is then computed.
#
# Details --
#
# Different values of "thresh" will be used in separate runs. Graphs will be drawn for the
# different "thresh" values, showing the rejection rates and the R-squares of dbMEM and
# Mantel analyses as a function of the autocorrelation range values.
#
# Parameters of the function --
#
# nsim : number of surfaces to be simulated.
# spacing = {1,2,3,4,5} for size of simulation surface = {20,29,38,47,56}.
#       This is also the truncation threshold used by function PCNM.
# range : range of the variogram for generation of SA on the surface.
# thresh: truncation distance for Mantel tests; {5,10,15,20} in our simulations.
# maxD : distance to use when geo.D[geo.D > thresh] <- maxD.
#       For spacing = 5, maxD = 63.63961.
# mean : mean of the simulated data.
# nugget: nugget parameter.
# var : total variance of the simulated data; var = nugget + sill.
# nperm1 : number of permutations for dbMEM analysis.
# nperm2 : number of permutations for Mantel analysis.
#
# Simulation of Gaussian random fields controlled by a spherical variogram:
# R.option=2 : With R 2.15, use function GaussRF() of RandomFields 2.0.66
# R.option=3 : With R 3.0, use function RFsimulate() of RandomFields 3.0.10
#
# Permutational test in dbMEM analysis: see function R2.test.perm().
#
# Value (output list) --
#
# rej.dbMEM : rejection rates, dbMEM analysis, parametric and permutation tests.
# rej.Mantel : rejection rates, Mantel analysis, permutation test.
# R.dbMEM : R-square & adj. R-square of indiv. simulations dbMEM anal. (matrix).
# p.dbMEM : p-values (param. & perm.) of individual simul. dbMEM anal. (matrix).
# r.Mantel : Mantel correlation of individual simulations (vector) with geoD.
# p.Mantel : p-values of individual simulations, Mantel analysis (vector).
# param : vector listing the run parameters: {nsim,spacing,range,nperm1,nperm2}.
# geo.D : geographic distance matrix for the grid sample, shown only if nsim=1.
# Y.D : response distance matrix for the grid sample, shown only when nsim=1.
# Surface : matrix with the whole response surface, shown only when nsim=1.
#       A map can be plotted using image(x$Surface)
#
# References --
#
# Schlather, M., P. Menck, R. Singleton, B. Pfaff and R Core team (2013).
#   RandomFields: Simulation and analysis of random fields.
#   R package version 2.0.66. http://CRAN.R-project.org/package=RandomFields
#
# Schlather, M., A. Malinowski, M. Oesting, D. Boecker, K. Strokorb, S. Engelke,
#   J. Martini, P. Menck, S. Gross, K. Burmeister, J. Manitz, R. Singleton,
#   B. Pfaff and R Core Team (2014). RandomFields: Simulation and Analysis
```



```
# of Random Fields. R package version 3.0.10.
# http://CRAN.R-project.org/package=RandomFields
#
# License: GPL-2
# Authors: Pierre Legendre and D. Borcard, July 2014
{
require(RandomFields)
require(vegan)
require(PCNM)
param <-c(nsim=nsim, spacing=spacing, range=range, thresh=thresh, nperm1=nperm1,
nperm2=nperm2)
#
if(R.option==2) {
  cat("Function GaussRF() of RandomFields 2.0.66 is used with R 2.15\n")
} else if(R.option==3) {
  cat("Function RFsimulate() of RandomFields 3.0.10 is used with R 3.0\n")
  RFOptions(spConform=FALSE)
  RFOptions(seed=NULL)
} else { stop("Error in R.option parameter") }
#
# Set up the simulation surface and the sampling grid parameters
size <- 9*spacing+11 # size = {20,29,38,47,56} depending on "spacing"
coo <- 1:size # Coordinates of points on simulation surface in each direction
#
# Coordinates of the sample of n=100 points on the surface
grid.coord <- expand.grid(seq(6,(size-5),spacing), seq(6,(size-5),spacing))
n <- nrow(grid.coord)
geo.D <- dist(grid.coord)
# cat("maxD =",maxD,"\n")
if(is.null(maxD)) maxD <- max(geo.D)
cat("maxD =",maxD,"\n")
#
# Construct dbMEM eigenfunctions
dbMEM.grid <- PCNM(geo.D,thresh=spacing,dbMEM=TRUE)
dbMEM <- as.data.frame(dbMEM.grid$vectors)
Q <- qr(as.matrix(dbMEM))
# Truncate geographic matrix D for Mantel test
geo.D[geo.D > thresh] <- maxD
# Prepare matrix and vectors to receive the simulation results
surf <- vector(mode="numeric", length=n)
R.dbMEM <- matrix(NA,nsim,2)
colnames(R.dbMEM) <- c("R.square","adj.R.square")
p.dbMEM <- matrix(NA,nsim,2)
colnames(p.dbMEM) <- c("Parametric.P","Permutational.P")
r.Mantel <- vector(mode="numeric", length=nsim)
p.Mantel <- vector(mode="numeric", length=nsim)
###
# Main simulation loop
a <- system.time({ # How much time for the simulations?
#
for(kk in 1:nsim) {
  if(range==0) {
    surf <- rnorm(n)
  } else {
    ## Simulate autocorrelated surface with spherical variogram model
    if(R.option==2) {
      Surface <- GaussRF(coo, coo, model="spherical", grid=TRUE,
        param=c(mean=mean, var=var, nugget=nugget, scale=range))
    }
    if(R.option==3) {
      model = RMspheric(var=var, scale=range)
      Surface <- RFsimulate(model = model, x=coo, y=coo, grid=TRUE)
      ### DB: if a nugget is sought, argument err.model in RFsimulate
      ### can be used: err.model=RMnugget(var=var)
    }
  }
}
```

```
    }
  #
  for(k in 1:n) surf[k] <- Surface[grid.coord[k,2], grid.coord[k,1]]
  }
  Y.D <- dist(surf)
  #
  # dbMEM analysis. No selection of the eigenfunctions
  lm.res <- lm(surf ~ ., data=dbMEM)
  R.dbMEM[kk,] <- c(summary(lm.res)$r.squared, summary(lm.res)$adj.r.squared)
  F.vec <- summary(lm.res)$fstatistic
  # Parametric test results (for the time being)
  p.dbMEM[kk,1] <- pf(F.vec[1], F.vec[2], F.vec[3], lower.tail=FALSE)
  if(nperm1==0) { p.dbMEM[kk,2] <- NA } else {
    p.dbMEM[kk,2] <- R2.test.perm(surf, dbMEM, nperm=nperm1, Q=Q, dbMEM=TRUE)$P }
  # Mantel test using vegan's mantel() function
  res.Mantel <- mantel(Y.D, geo.D, permutations=nperm2)
  r.Mantel[kk] <- res.Mantel$statistic
  p.Mantel[kk] <- res.Mantel$signif
  }
  rej.dbMEM <-
    c(length(which(p.dbMEM[,1]<=0.05)), length(which(p.dbMEM[,2]<=0.05)))/nsim
  names(rej.dbMEM) <- c("p.param", "p.perm")
  rej.Mantel <- length(which(p.Mantel <= 0.05))/nsim
  #
  })
  a[3] <- sprintf("%2f", a[3])
  cat("Simulation time =", a[3], " sec", '\n')
  #
  # Save Surface, geo.D and Y.D only if a single simulation has been produced
  if(nsim==1) { Surface <- Surface; geo.D <- geo.D; Y.D <- Y.D }
  else { Surface <- NULL; geo.D <- NULL; Y.D <- NULL }
  #
  out <- list(rej.dbMEM=rej.dbMEM, rej.Mantel=rej.Mantel, R.dbMEM=R.dbMEM, p.dbMEM=p.dbMEM,
    r.Mantel=r.Mantel, p.Mantel=p.Mantel, param=param, geo.D=geo.D, Y.D=Y.D, Surface=Surface)
  }

R2.test.perm <- function(Y, X, nperm=999, Q=NULL, dbMEM=FALSE)
# Permutation test for R2 statistic in regression or RDA
# Some operations on X=dbMEM are transferred to the main function LFB.simul1()
# in the case of dbMEM analysis.
#
# Parameters of the function --
# Y : Matrix or vector of response data.
# X : Matrix of explanatory data, e.g. a file of dbMEM.
# nperm : Number of permutations for the test.
# dbMEM=FALSE: Normal computation of statistics for regression or RDA:
#               Centre X, QR decomposition of X, compute adjusted R-square.
# dbMEM=TRUE : Do not centre X since the matrix of dbMEM is centred.
#               Do not compute the adjusted R-square.
# License: GPL-2
# Author:: Pierre Legendre, June 2014
{
  Y.c <- scale(Y, center=TRUE, scale=FALSE) # Centre Y
  SS.Y <- sum(Y.c^2)
  X <- as.matrix(X)
  n <- nrow(X)
  if(!dbMEM) {
    X <- scale(X, center=TRUE, scale=FALSE) # Centre X
    Q <- qr(X) # QR decomposition of X
  }
  Yfit.X <- qr.fitted(Q, Y.c)
  # Compute statistics
  Rsquare <- sum(Yfit.X^2)/SS.Y
  m <- Q$rank
```

---

```
residualDF <- n-m-1
F <- (Rsquare*residualDF)/((1-Rsquare)*m)
if(dbMEM) { adjRsqr <- NA } else {
  totalDF <- n-1
  adjRsqr <- 1-((1-Rsquare)*totalDF/residualDF)
}
# Permutation test of F
if(nperm > 0) {
  nPGE <- 1
  for(i in 1:nperm) {
    YfitPerm <- qr.fitted(Q, sample(Y.c))
    RsquarePerm <- sum(YfitPerm^2)/SS.Y
    FPerm <- (RsquarePerm*residualDF)/((1-RsquarePerm)*m)
    if(FPerm >= F) nPGE <- nPGE+1
  }
  P <- nPGE/(nperm+1)
} else { P <- NA }
#
if(dbMEM) { out <- list(P=P) }
else {out <- list(Rsquare=Rsquare, F=F, P=P, adjRsqr=adjRsqr, nperm=nperm, m=m,
residualDF=residualDF) }
}
```

File **run.LFB.simul2.batch1.txt** — Batch run for series 2 simulations, thresh=5

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul2.R"  
source("LFB.simul2.R")
```

```
res2.5.0.5 = LFB.simul2(nsim=1000, spacing=5, range=0, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.5.5 = LFB.simul2(nsim=1000, spacing=5, range=5, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.10.5 = LFB.simul2(nsim=1000, spacing=5, range=10, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.15.5 = LFB.simul2(nsim=1000, spacing=5, range=15, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.20.5 = LFB.simul2(nsim=1000, spacing=5, range=20, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.25.5 = LFB.simul2(nsim=1000, spacing=5, range=25, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.30.5 = LFB.simul2(nsim=1000, spacing=5, range=30, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.35.5 = LFB.simul2(nsim=1000, spacing=5, range=35, thresh=5, nperm1=0, nperm2=999)  
#  
res2.5.40.5 = LFB.simul2(nsim=1000, spacing=5, range=40, thresh=5, nperm1=0, nperm2=999)  
#  
save(res2.5.0.5,res2.5.5.5,res2.5.10.5,res2.5.15.5,res2.5.20.5,res2.5.25.5,res2.5.30.5,res2.5.35.5,  
res2.5.40.5, file="simul2,spacing=5,thresh=5.RData")
```

---

File **run.LFB.simul2.batch2.txt** — Batch run for series 2 simulations, thresh=10

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul2.R"  
source("LFB.simul2.R")
```

```
res2.5.0.10 = LFB.simul2(nsim=1000, spacing=5, range=0, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.5.10 = LFB.simul2(nsim=1000, spacing=5, range=5, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.10.10 = LFB.simul2(nsim=1000, spacing=5, range=10, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.15.10 = LFB.simul2(nsim=1000, spacing=5, range=15, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.20.10 = LFB.simul2(nsim=1000, spacing=5, range=20, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.25.10 = LFB.simul2(nsim=1000, spacing=5, range=25, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.30.10 = LFB.simul2(nsim=1000, spacing=5, range=30, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.35.10 = LFB.simul2(nsim=1000, spacing=5, range=35, thresh=10, nperm1=0, nperm2=999)  
#  
res2.5.40.10 = LFB.simul2(nsim=1000, spacing=5, range=40, thresh=10, nperm1=0, nperm2=999)  
#  
save(res2.5.0.10, res2.5.5.10, res2.5.10.10, res2.5.15.10, res2.5.20.10, res2.5.25.10, res2.5.30.10,  
res2.5.35.10, res2.5.40.10, file="simul2,spacing=5,thresh=10.RData")
```

---

File **run.LFB.simul2.batch3.txt** — Batch run for series 2 simulations, thresh=15

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul2.R"  
source("LFB.simul2.R")
```

```
res2.5.0.15 = LFB.simul2(nsim=1000, spacing=5, range=0, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.5.15 = LFB.simul2(nsim=1000, spacing=5, range=5, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.10.15 = LFB.simul2(nsim=1000, spacing=5, range=10, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.15.15 = LFB.simul2(nsim=1000, spacing=5, range=15, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.20.15 = LFB.simul2(nsim=1000, spacing=5, range=20, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.25.15 = LFB.simul2(nsim=1000, spacing=5, range=25, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.30.15 = LFB.simul2(nsim=1000, spacing=5, range=30, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.35.15 = LFB.simul2(nsim=1000, spacing=5, range=35, thresh=15, nperm1=0, nperm2=999)  
#  
res2.5.40.15 = LFB.simul2(nsim=1000, spacing=5, range=40, thresh=15, nperm1=0, nperm2=999)  
#  
save(res2.5.0.15, res2.5.5.15, res2.5.10.15, res2.5.15.15, res2.5.20.15, res2.5.25.15, res2.5.30.15,  
res2.5.35.15, res2.5.40.15, file="simul2,spacing=5,thresh=15.RData")
```

---

File **run.LFB.simul2.batch4.txt** — Batch run for series 2 simulations, thresh=20

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul2.R"  
source("LFB.simul2.R")
```

```
res2.5.0.20 = LFB.simul2(nsim=1000, spacing=5, range=0, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.5.20 = LFB.simul2(nsim=1000, spacing=5, range=5, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.10.20 = LFB.simul2(nsim=1000, spacing=5, range=10, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.15.20 = LFB.simul2(nsim=1000, spacing=5, range=15, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.20.20 = LFB.simul2(nsim=1000, spacing=5, range=20, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.25.20 = LFB.simul2(nsim=1000, spacing=5, range=25, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.30.20 = LFB.simul2(nsim=1000, spacing=5, range=30, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.35.20 = LFB.simul2(nsim=1000, spacing=5, range=35, thresh=20, nperm1=0, nperm2=999)  
#  
res2.5.40.20 = LFB.simul2(nsim=1000, spacing=5, range=40, thresh=20, nperm1=0, nperm2=999)  
#  
save(res2.5.0.20, res2.5.5.20, res2.5.10.20, res2.5.15.20, res2.5.20.20, res2.5.25.20, res2.5.30.20,  
res2.5.35.20, res2.5.40.20, file="simul2,spacing=5,thresh=20.RData")
```

---

```
LFB.simul3 <- function(nsim=100, spacing=5, range=10, N=50, thresh=NULL, maxD=NULL,
new.Del=TRUE, mean=0.0, nugget=0.0, var=1, nperm1=99, nperm2=99, R.option=3)
#
# Task: Simulation function for the Legendre, Fortin & Borcard paper.
# Map sizes are chosen to harbour 100 sampling units (n=100) with spacing of
# {1,2,3,4,5} with a buffer zone of 5 pixels all around the sampled area.
#
# In the LFB.simul3 simulations, autocorrelated surfaces are generated with different
# range values, as in the LFB.simul1 function. In this function, however, N points are
# selected at random from among the 100 points of the regular sampling grid. A Delaunay
# triangulation is constructed among these points. New distances are computed among these
# N points; the distance is the minimum number of Delaunay segments separating two points.
#
# Details --
#
# For the N selected points, a Delaunay triangulation is computed, as well as a matrix of
# dbMEM eigenfunctions. If "thresh" is not NULL, the matrix of Delaunay graph distances is
# truncated, which means that any value larger than a selected threshold is replaced by
# the largest distance actually found in the matrix, max.D. The simulated response values
# for the N selected points are kept in a vector. The Mantel test is computed for the
# truncated matrix of Delaunay graph distances.
#
# Different values of "thresh" will be used in separate runs. Graphs will be drawn for the
# different "thresh" values, showing the rejection rates and the R-squares of dbMEM and
# Mantel analyses as a function of the autocorrelation range values.
#
# Permutation test in dbMEM analysis are carried out by function R2.test.perm.
#
# Simulation of Gaussian random fields controlled by a spherical variogram:
# R.option=2 : With R 2.15, use function GaussRF() of RandomFields 2.0.66
# R.option=3 : With R 3.0, use function RFsimulate() of RandomFields 3.0.10
#
# Parameters of the function --
#
# nsim : number of surfaces to be simulated.
# spacing = {1,2,3,4,5} for size of simulation surface = {20,29,38,47,56}.
# range : range of the variogram for generation of SA on the surface.
# N : Number of points to sample on the regular grid.
# thresh: truncation D of Delaunay triangulation for Mantel tests.
# maxD : distance to use when geo.D[geo.D > thresh] --
#         Use the maximum distance on the grid if maxD=NULL: sqrt(9^2*2)*5
#         Use the user-defined value of maxD if maxD is not NULL.
# new.Del=TRUE : New Delaunay triangulation and MEM file for every SA surface.
#               =FALSE: A single Delaunay triangulation and MEM file for all simulations.
# mean : mean of the simulated data
# nugget: nugget parameter
# var : total variance of the simulated data; var = nugget + sill (PL check).
# nperm1 : number of permutations for dbMEM analysis.
# nperm2 : number of permutations for Mantel analysis.
#
# Value (output list) --
#
# rej.dbMEM : rejection rates, dbMEM analysis, parametric and permutation tests.
# rej.Mantel : rejection rates, Mantel analysis, permutation test.
# R.dbMEM : R-square & adj. R-square of indiv. simulations dbMEM anal. (matrix).
# p.dbMEM : p-values (param. & perm.) of individual simul. dbMEM anal. (matrix).
# r.Mantel : Mantel correlations of individual simulations (vector) with geoD.
# p.Mantel : p-values of individual simulations, Mantel analysis (vector).
# param : vector of run parameters: {nsim,spacing,range,thresh,nperm1,nperm2}.
# Del.D : Delaunay distance matrix for the grid sample, shown only if nsim=1.
# Y.D : response distance matrix for the grid sample, shown only when nsim=1.
# Surface : matrix with the whole response surface, shown only when nsim=1.
#           A map can be plotted using image(x$Surface)
```



```
#
# References --
#
# Schlather, M., P. Menck, R. Singleton, B. Pfaff and R Core team (2013).
#   RandomFields: Simulation and analysis of random fields.
#   R package version 2.0.66. http://CRAN.R-project.org/package=RandomFields
#
# Schlather, M., A. Malinowski, M. Oesting, D. Boecker, K. Strokorb, S. Engelke,
#   J. Martini, P. Menck, S. Gross, K. Burmeister, J. Manitz, R. Singleton,
#   B. Pfaff and R Core Team (2014). RandomFields: Simulation and Analysis
#   of Random Fields. R package version 3.0.10.
#   http://CRAN.R-project.org/package=RandomFields
#
# License: GPL-2
# Authors: Pierre Legendre and D. Borcard, July 2014
{
require(spatstat)
require(RandomFields)
require(vegan)
require(PCNM)
param <-c(nsim=nsim, spacing=spacing, range=range, thresh=thresh, nperm1=nperm1,
nperm2=nperm2)
#
if(R.option==2) {
  cat("Function GaussRF() of RandomFields 2.0.66 is used with R 2.15\n")
} else if(R.option==3) {
  cat("Function RFsimulate() of RandomFields 3.0.10 is used with R 3.0\n")
  RFoptions(spConform=FALSE)
  RFoptions(seed=NULL)
} else { stop("Error in R.option parameter") }
#
# Set up the simulation surface and the sampling grid parameters
size <- 9*spacing+11      # size = {20,29,38,47,56} depending on "spacing"
coo <- 1:size            # Coordinates of points on simulation surface in each direction
#
# Coordinates of the sample of n=100 points on the surface
grid.coord <- expand.grid(seq(6,(size-5),spacing), seq(6,(size-5),spacing))
n <- nrow(grid.coord)
if(N > n) stop("N is larger than the number of points on the sample grid")
#
if(!new.Del) { # Use the same Delaunay triangulation and MEM file for all surfaces
  # Select N points for Delaunay triangulation
  vec.N = sort(sample(n, N))      # Selected N points for Delaunay triangulation
  print(vec.N)
  cat(vec.N, "\n")
  coord.N = grid.coord[vec.N,] # Coordinates of the selected points
  # Delaunay triangulation -- D matrix in number of steps along the triangulation
  Del.D <- as.dist(delaunay.distance(ppp(coord.N[,1],coord.N[,2],xrange=c(1,size),
                                     xrange=c(1,size))))
  if(is.null(maxD)) maxD. <- max(Del.D)
  cat("new.Del=FALSE: maxD. in Delaunay =",maxD., "\n")
  # Truncate the Delaunay matrix D that will be used for the Mantel test
  Del.D[Del.D > thresh] <- maxD.
  # Construct dbMEM eigenfunctions
  geo.D <- dist(coord.N)
  dbMEM <- PCNM(geo.D,dbMEM=TRUE,silent=TRUE)
  dbMEM <- as.data.frame(dbMEM.N$vectors)
  Q <- qr(as.matrix(dbMEM))
}
#
# Prepare matrix and vectors to receive the simulation results
surf      <- vector(mode="numeric", length=N)
R.dbMEM    <- matrix(NA,nsim,2)
colnames(R.dbMEM) <- c("R.square", "adj.R.square")
```

```
p.dbMEM <- matrix(NA,nsim,2)
colnames(p.dbMEM) <- c("Parametric.P","Permutational.P")
r.Mantel <- vector(mode="numeric", length=nsim)
p.Mantel <- vector(mode="numeric", length=nsim)
MaxD <- 0
###
# Main simulation loop
a <- system.time({          # How much time for the simulations?
#
for(kk in 1:nsim) {
  if(new.Del) { # New Delaunay triangulation and MEM file for every SA surface
    repeat { ### try() : procedure to go around a bug in package deldir
      vec.N = sort(sample(n, N)) # Selected points for Delaunay triangulation
      coord.N = grid.coord[vec.N,] # Coordinates of the selected points
      tmp <- ppp(coord.N$Var1/size,coord.N$Var2/size)
      # if(!is.ppp(tmp)) cat(kk,"No\n") else cat(kk,"Yes\n")
      tst = try(Del.D <- delaunay.distance(tmp), silent=TRUE)
      if(class(tst)=="matrix") {
        break } else { cat("kk =",kk,"### Error in delaunay.dist()\n") }
    }

    Del.D <- as.dist(Del.D)
    if(is.null(maxD)) maxD. <- max(Del.D)
    if(maxD. > MaxD) MaxD <- maxD.
    # cat(kk,"- maxD. =",maxD.,"\n")
    # Truncate the Delaunay matrix D that will be used for the Mantel test
    Del.D[Del.D > thresh] <- maxD.
    # Construct dbMEM eigenfunctions
    geo.D <- dist(coord.N)
    dbMEM.N <- PCNM(geo.D,dbMEM=TRUE,silent=TRUE)
    dbMEM <- as.data.frame(dbMEM.N$vectors)
    Q <- qr(as.matrix(dbMEM))
  }
  if(range==0) {
    surf <- rnorm(N)
  } else {
    ## Simulate autocorrelated surface with spherical variogram model
    if(R.option==2) {
      Surface <- GaussRF(coo, coo, model="spherical", grid=TRUE,
        param=c(mean=mean, var=var, nugget=nugget, scale=range))
    }
    if(R.option==3) {
      model = RMspheric(var=var, scale=range)
      Surface <- RFsimulate(model = model, x=coo, y=coo, grid=TRUE)
      ### DB: if a nugget is sought, argument err.model in RFsimulate
      ### can be used: err.model=RMnugget(var=var)
    }
  }
  #
  for(k in 1:N) surf[k] <- Surface[coord.N[k,2], coord.N[k,1]]
}
if(nsim==1) return(list(coord.N=coord.N, surf=surf, Del.D=Del.D, dbMEM=dbMEM, N=N))
Y.D <- dist(surf)
#
# dbMEM analysis. No selection of the eigenfunctions
lm.res <- lm(surf ~ .,data=dbMEM)
R.dbMEM[kk,] <- c(summary(lm.res)$r.squared, summary(lm.res)$adj.r.squared)
F.vec <- summary(lm.res)$fstatistic
# Parametric test results (for the time being)
p.dbMEM[kk,1] <- pf(F.vec[1], F.vec[2], F.vec[3], lower.tail=FALSE)
if(nperm1==0) { p.dbMEM[kk,2] <- NA } else {
  p.dbMEM[kk,2] <- R2.test.perm(surf,dbMEM,nperm=nperm1,Q=Q, dbMEM=TRUE)$P }
# Mantel test using vegan's mantel() function
res.Mantel <- mantel(Y.D, Del.D, permutations=nperm2)
r.Mantel[kk] <- res.Mantel$statistic
```

```
p.Mantel[kk] <- res.Mantel$signif
}
rej.dbMEM <-
  c(length(which(p.dbMEM[,1]<=0.05)),length(which(p.dbMEM[,2]<=0.05)))/nsim
names(rej.dbMEM) <- c("p.param", "p.perm")
rej.Mantel <- length(which(p.Mantel <= 0.05))/nsim
if(new.Del) cat("new.Del=TRUE: MaxD in Delaunay =",MaxD,"\n")
#
})
a[3] <- sprintf("%2f",a[3])
cat("\n","Simulation time =",a[3]," sec","\n')
#
# Save Surface, Del.D and Y.D only if a single simulation has been produced
if(nsim==1) {Surface<-Surface; Del.D<-Del.D; Y.D<-Y.D}
  else {Surface<-NULL; Del.D<-NULL; Y.D<-NULL}
#
out <- list(rej.dbMEM=rej.dbMEM, rej.Mantel=rej.Mantel, R.dbMEM=R.dbMEM, p.dbMEM=p.dbMEM,
r.Mantel=r.Mantel, p.Mantel=p.Mantel, param=param, Del.D=Del.D, Y.D=Y.D, Surface=Surface)
}

R2.test.perm <- function(Y, X, nperm=999, Q=NULL, dbMEM=FALSE)
# Permutation test for R2 statistic in regression or RDA
# Some operations on X=dbMEM are transferred to the main function LFB.simul1()
# in the case of dbMEM analysis.
#
# Parameters of the function --
# Y : Matrix or vector of response data.
# X : Matrix of explanatory data, e.g. a file of dbMEM.
# nperm : Number of permutations for the test.
# dbMEM=FALSE: Normal computation of statistics for regression or RDA:
#             Centre X, QR decomposition of X, compute adjusted R-square.
# dbMEM=TRUE : Do not centre X since the matrix of dbMEM is centred.
#             Do not compute the adjusted R-square.
# License: GPL-2
# Author:: Pierre Legendre, June 2014
{
Y.c <- scale(Y,center=TRUE,scale=FALSE)    # Centre Y
SS.Y <- sum(Y.c^2)
X <- as.matrix(X)
n <- nrow(X)
if(!dbMEM) {
  X <- scale(X,center=TRUE,scale=FALSE)    # Centre X
  Q <- qr(X)                               # QR decomposition of X
}
Yfit.X <- qr.fitted(Q, Y.c)
# Compute statistics
Rsquare <- sum(Yfit.X^2)/SS.Y
m <- Q$rank
residualDF <- n-m-1
F <- (Rsquare*residualDF)/((1-Rsquare)*m)
if(dbMEM) { adjRsqr <- NA } else {
  totalDF <- n-1
  adjRsqr <- 1-((1-Rsquare)*totalDF/residualDF)
}
# Permutation test of F
if(nperm > 0) {
  nPGE <- 1
  for(i in 1:nperm) {
    YfitPerm <- qr.fitted(Q, sample(Y.c))
    RsquarePerm <- sum(YfitPerm^2)/SS.Y
    FPerm <- (RsquarePerm*residualDF)/((1-RsquarePerm)*m)
    if(FPerm >= F) nPGE <- nPGE+1
  }
  P <- nPGE/(nperm+1)
```

```
    } else { P <- NA }  
#  
if(dbMEM) { out <- list(P=P) }  
else {out <- list(Rsquare=Rsquare, F=F, P=P, adjRsqr=adjRsqr, nperm=nperm, m=m,  
residualDF=residualDF) }  
}
```

File **run.LFB.simul3.batch1.txt** — Batch run for series 3 simulations, thresh=1

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul3.R"  
source("LFB.simul3.R")
```

```
res3.5.0.1 = LFB.simul3(nsim=1000, spacing=5, range=0, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.5.1 = LFB.simul3(nsim=1000, spacing=5, range=5, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.10.1 = LFB.simul3(nsim=1000, spacing=5, range=10, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.15.1 = LFB.simul3(nsim=1000, spacing=5, range=15, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.20.1 = LFB.simul3(nsim=1000, spacing=5, range=20, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.25.1 = LFB.simul3(nsim=1000, spacing=5, range=25, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.30.1 = LFB.simul3(nsim=1000, spacing=5, range=30, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.35.1 = LFB.simul3(nsim=1000, spacing=5, range=35, thresh=1, nperm1=999, nperm2=999)  
#  
res3.5.40.1 = LFB.simul3(nsim=1000, spacing=5, range=40, thresh=1, nperm1=999, nperm2=999)  
#  
save(res3.5.0.1, res3.5.5.1, res3.5.10.1, res3.5.15.1, res3.5.20.1, res3.5.25.1, res3.5.30.1, res3.5.35.1,  
res3.5.40.1, file="simul3,spacing=5,thresh=1.RData")
```

---

File **run.LFB.simul3.batch2.txt** — Batch run for series 3 simulations, thresh=3

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul3.R"  
source("LFB.simul3.R")
```

```
res3.5.0.3 = LFB.simul3(nsim=1000, spacing=5, range=0, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.5.3 = LFB.simul3(nsim=1000, spacing=5, range=5, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.10.3 = LFB.simul3(nsim=1000, spacing=5, range=10, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.15.3 = LFB.simul3(nsim=1000, spacing=5, range=15, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.20.3 = LFB.simul3(nsim=1000, spacing=5, range=20, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.25.3 = LFB.simul3(nsim=1000, spacing=5, range=25, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.30.3 = LFB.simul3(nsim=1000, spacing=5, range=30, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.35.3 = LFB.simul3(nsim=1000, spacing=5, range=35, thresh=3, nperm1=999, nperm2=999)  
#  
res3.5.40.3 = LFB.simul3(nsim=1000, spacing=5, range=40, thresh=3, nperm1=999, nperm2=999)  
#  
save(res3.5.0.3, res3.5.5.3, res3.5.10.3, res3.5.15.3, res3.5.20.3, res3.5.25.3, res3.5.30.3, res3.5.35.3,  
res3.5.40.3, file="simul3,spacing=5,thresh=3.RData")
```

---

File **run.LFB.simul3.batch3.txt** — Batch run for series 3 simulations, thresh=5

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul3.R"  
source("LFB.simul3.R")
```

```
res3.5.0.5 = LFB.simul3(nsim=1000, spacing=5, range=0, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.5.5 = LFB.simul3(nsim=1000, spacing=5, range=5, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.10.5 = LFB.simul3(nsim=1000, spacing=5, range=10, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.15.5 = LFB.simul3(nsim=1000, spacing=5, range=15, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.20.5 = LFB.simul3(nsim=1000, spacing=5, range=20, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.25.5 = LFB.simul3(nsim=1000, spacing=5, range=25, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.30.5 = LFB.simul3(nsim=1000, spacing=5, range=30, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.35.5 = LFB.simul3(nsim=1000, spacing=5, range=35, thresh=5, nperm1=999, nperm2=999)  
#  
res3.5.40.5 = LFB.simul3(nsim=1000, spacing=5, range=40, thresh=5, nperm1=999, nperm2=999)  
#  
save(res3.5.0.5, res3.5.5.5, res3.5.10.5, res3.5.15.5, res3.5.20.5, res3.5.25.5, res3.5.30.5, res3.5.35.5,  
res3.5.40.5, file="simul3,spacing=5,thresh=5.RData")
```

---

File **run.LFB.simul3.batch4.txt** — Batch run for series 3 simulations, thresh=10

```
# Load the necessary R packages
```

```
# Source the function for calculation, file "LFB.simul3.R"  
source("LFB.simul3.R")
```

```
res3.5.0.10 = LFB.simul3(nsim=1000, spacing=5, range=0, thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.5.10 = LFB.simul3(nsim=1000, spacing=5, range=5, thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.10.10 =LFB.simul3(nsim=1000,spacing=5,range=10,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.15.10 =LFB.simul3(nsim=1000,spacing=5,range=15,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.20.10 =LFB.simul3(nsim=1000,spacing=5,range=20,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.25.10 =LFB.simul3(nsim=1000,spacing=5,range=25,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.30.10 =LFB.simul3(nsim=1000,spacing=5,range=30,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.35.10 =LFB.simul3(nsim=1000,spacing=5,range=35,thresh=10,nperm1=999,nperm2=999)  
#  
res3.5.40.10 =LFB.simul3(nsim=1000,spacing=5,range=40,thresh=10,nperm1=999,nperm2=999)  
#  
save(res3.5.0.10,res3.5.5.10,res3.5.10.10,res3.5.15.10,res3.5.20.10,res3.5.25.10,res3.5.30.10,  
res3.5.35.10,res3.5.40.10, file="simul3,spacing=5,thresh=10.RData")
```

---