Methods in Ecology and Evolution

British Ecological Society

Methods in Ecology and Evolution 2011, **2**, 269–277 doi: 10.1111/j.2041-210X.2010.00078.x

Testing the significance of canonical axes in redundancy analysis

Pierre Legendre[1]*, Jari Oksanen[2] and Cajo J. F. ter Braak[3]

[1]Département de Sciences Biologiques, Université de Montréal, C.P. 6128, Succursale Centre-Ville, Montréal, QC H3C 3J7, Canada; [2]Department of Biology, University of Oulu, P.O. Box 3000, FIN-90014 Oulu, Finland and [3]Biometris, Wageningen University and Research Centre, Box 100, 6700 AC Wageningen, The Netherlands

Summary

1. Tests of significance of the individual canonical axes in redundancy analysis allow researchers to determine which of the axes represent variation that can be distinguished from random. Variation along the significant axes can be mapped, used to draw biplots or interpreted through subsequent analyses, whilst the nonsignificant axes may be dropped from further consideration.

2. Three methods have been implemented in computer programs to test the significance of the canonical axes; they are compared in this paper. The simultaneous test of all individual canonical axes, which is appealing because of its simplicity, produced incorrect (highly inflated) levels of type I error for the axes following those corresponding to true relationships in the data, so it is invalid. The 'marginal' testing method implemented in the 'vegan' R package and the 'forward' testing method implemented in the program CANOCO were found to have correct levels of type I error and comparable power. Permutation of the residuals achieved greater power than permutation of the raw data.

3. R functions found in a Supplement to this paper provide the first formal description of the 'marginal' and 'forward' testing methods.

Key-words: canonical redundancy analysis (RDA), numerical simulations, power, tests of significance, type I error

Introduction

Redundancy analysis (RDA, Rao 1964) and canonical correspondence analysis (CCA, ter Braak 1986, 1987) are two forms of asymmetric canonical analysis widely used by ecologists and palaeoecologists. 'Asymmetric' means that the two data matrices used in the analysis do not play the same role: there is a matrix of response variables, denoted **Y**, which often contains community composition data, and a matrix of explanatory variables (e.g. environmental), denoted **X**, which is used to explain the variation in **Y**, as in regression analysis. Contrast this with canonical correlation and co-inertia analyses where the two matrices play the same role in the analysis and can be interchanged; see, however, Tso (1981) for an asymmetric interpretation of canonical correlation analysis. RDA and CCA produce ordinations of **Y** constrained by **X**.

This paper deals with methods to test the significance of the canonical axes that emerge from this type of analysis. The canonical axes are those that are formed by linear combinations of the predictor variables; they are sometimes referred to as 'constrained axes'. The section 'Background: the algebra of redundancy analysis' will show how they are computed. Individual canonical axes may be tested when the overall relationship (R^2) between **Y** and **X** has been shown to be significant. We will concentrate on RDA; our conjecture is that the conclusions derived from our simulations should apply to CCA as well.

As we deal with complex, multivariate data influenced by many factors, it is to be expected that several independent structures coexist in the response data. If these structures are linearly independent, they should appear on different canonical axes. Each one should be identifiable by a test of significance. Canonical axes that explain no more variation than random should also be detected; they do not need to be further considered in the interpretation of the results.

It is not always necessary to test the significance of the canonical axes when there are only a few. However, researchers often face situations where there is a large number of canonical axes; they may want to know how many canonical axes should be examined, plotted, and interpreted. Spatial modelling is a good example of these situations: when analysing the spatial variation of species-rich communities (hundreds of species, hundreds of sites) by spatial eigenfunctions, we may end up with hundreds of canonical axes. Different types of spatial eigenfunctions have been described in recent years to model

*Correspondence author. E-mail: Pierre.Legendre@umontreal.ca
Correspondence site: http://www.respond2articles.com/MEE/

© 2010 The Authors. Methods in Ecology and Evolution © 2010 British Ecological Society

the spatial structure of multivariate response data: Griffith's (2000) spatial eigenfunctions from a connection matrix of neighbouring regions or sites; Moran's eigenvector maps (MEM, Borcard & Legendre 2002; Borcard *et al.* 2004; Dray, Legendre, & Peres-Neto 2006); asymmetric eigenvector maps (AEM, Blanchet, Legendre, & Borcard 2008). It is then of interest to determine which of the canonical axes derived from these eigenfunctions represent variation that is more structured than random. This is the objective of the tests of significance of the canonical axes. Variation along the (hopefully few) significant axes can be mapped, used to draw biplots or interpreted through subsequent analyses. The nonsignificant axes can be dropped because they do not represent variation more structured than random. The eigenvalue associated with each axis expresses the variance accounted for by the canonical axis; the fraction of the response variation that each axis represents is a useful supplementary criterion: axes that account for less than, say, 1% or 5% of the variation may not need to be further analysed even if they are statistically significant.

Three methods have been proposed to test the significance of canonical axes in RDA. We will compare them using simulated data to determine which ones, if any, have correct levels of type I error (defined in the section 'Simulation method'). We will also compare these methods and two permutation procedures in terms of power. This paper provides the first formal description of these methods.

Background: the algebra of redundancy analysis

Redundancy analysis (RDA, Rao 1964) of a response matrix **Y** (with *n* objects and *p* variables) by an explanatory matrix **X** (with *n* objects and *m* variables) consists of two steps (Legendre & Legendre 1998, section 11·1). In the algebraic description that follows, the columns of matrices **Y** and **X** have been centred to have means of 0.

- Step 1 is a multivariate regression of **Y** on **X**, which produces a matrix of fitted values $\hat{\mathbf{Y}}$ through the linear equation:

$$\hat{\mathbf{Y}} = \mathbf{X}[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{Y} \qquad \text{eqn 1}$$

This is equivalent to a series of multiple linear regressions of the individual variables of **Y** on **X**, calculation of the vectors of fitted values and binding these column vectors to form matrix $\hat{\mathbf{Y}}$.

- Step 2 is a principal component analysis of $\hat{\mathbf{Y}}$. This PCA produces the canonical eigenvalues and eigenvectors as well as the canonical axes (object ordination scores). This step is performed to obtain reduced-space ordination diagrams displaying the objects, response variables, and explanatory variables for the most important axes of the canonical relationship.

Like the fitted values of a multiple linear regression, the canonical axes (object ordination scores) are also linear combinations of the explanatory variables in **X**. These linear combinations are the defining properties of canonical axes in the presentation of RDA by ter Braak & Prentice (1988) and ter Braak (1995). The present paper focuses on the problem of determining which of the canonical axes are important enough to warrant consideration, plotting and detailed analysis.

PARTIAL RDA

The statistical test for identifying the significant axes requires the notion of a partial RDA, that is, RDA with additional explanatory variables, called covariables, assembled in matrix **W**. In partial RDA, the linear effects of the explanatory variables in **X** on the response variables in **Y** are adjusted for the effects of the covariables in **W**.

In multiple regression, we know that the partial regression of **y** by **X** in the presence of covariables **W** can be computed in two different ways (Legendre & Legendre 1998, section 10·3·5). One first computes the residuals of **y** on **W** (noted $\mathbf{y}_{\mathrm{res}|\mathbf{W}}$) and the residuals of **X** on **W** ($\mathbf{X}_{\mathrm{res}|\mathbf{W}}$). Then, one can either regress $\mathbf{y}_{\mathrm{res}|\mathbf{W}}$ on $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$, or regress **y** on $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$. The same partial regression coefficients and vector of fitted values are obtained in both cases. The R^2 of the first analysis is the partial R^2 whereas that of the second analysis is the semipartial R^2.

The same two approaches can be used for partial RDA. First, one computes the residuals of **Y** on **W** (noted $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$) and the residuals of **X** on **W** ($\mathbf{X}_{\mathrm{res}|\mathbf{W}}$). Then, one can compute either the RDA of $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$ by $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$ or the RDA of **Y** by $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$. The two approaches produce the same canonical eigenvalues, eigenvectors and axes and can be used to test the significance of the canonical axes. In partial RDA, the canonical axes are linear combination of the adjusted **X** variables, $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$, and are orthogonal to the covariables in **W**. The R^2 obtained in the first approach is the partial canonical R^2 whereas that of the second analysis is the semipartial canonical R^2; these two statistics are described in eqns 9 and 10 below.

STATISTICS, SIMPLE RDA

The first step of RDA already leads to informative statistics. With matrices **Y** and $\hat{\mathbf{Y}}$, one can compute the following statistics:

- The canonical R^2, which Miller & Farr (1971) called the *bi-multivariate redundancy statistic*, measures the strength of the linear relationship between **Y** and **X**:

$$R^2_{\mathbf{Y}|\mathbf{X}} = \frac{\mathrm{SS}(\hat{\mathbf{Y}})}{\mathrm{SS}(\mathbf{Y})} \qquad \text{eqn 2}$$

where $\mathrm{SS}(\hat{\mathbf{Y}})$ is the total sum-of-squares (or sum of squared deviations from the means) of $\hat{\mathbf{Y}}$ and SS(**Y**) is the total sum-of-squares of **Y**. Miller & Farr (1971) derived this equation as follows. They considered the case where the *p* variables of **Y** are standardized, forming matrix $\mathbf{Y}_{\mathrm{stand}}$. They computed a principal component analysis (PCA) of $\mathbf{Y}_{\mathrm{stand}}$, then a multiple regression of each principal component *j* on the explanatory matrix **X**. This multiple regression is noted $\mathrm{PC}_j|\mathbf{X}$ and its R^2 is $R^2_{\mathrm{PC}_j|\mathbf{X}}$. Miller and Farr established that

$$R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}} = \frac{\sum_{j=1}^{p} \lambda_j R^2_{\text{PC}_j|\mathbf{X}}}{p} \qquad \text{eqn 3}$$

The PC$_j$ are the principal components of $\mathbf{Y}_{\text{stand}}$ and the λ_j are the corresponding eigenvalues. The sum of the eigenvalues is p, the number of variables in $\mathbf{Y}_{\text{stand}}$, because the variables in $\mathbf{Y}_{\text{stand}}$ have been standardized. So $R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}}$, where $\mathbf{Y}_{\text{stand}|\mathbf{X}}$ denotes the multivariate regression of $\mathbf{Y}_{\text{stand}}$ on **X**, is a weighted mean of the coefficients of determination of the PC$_j$ regressed on matrix **X**, the weights being given by the proportion of the variance of $\mathbf{Y}_{\text{stand}}$ occupied by each principal component (i.e. the eigenvalues divided by p). The same value of $R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}}$ would be obtained by calculating the mean of the coefficients of determination of the standardized $\mathbf{Y}_{\text{stand}}$ variables regressed one by one on **X**:

$$R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}} = \frac{\sum_{j=1}^{p} R^2_{y'|\mathbf{X}}}{p} \qquad \text{eqn 4}$$

For the general case, where **Y** is not standardized, $R^2_{\mathbf{Y}|\mathbf{X}}$ is computed using eqn 2.

- The adjusted R^2 is computed using the classical Ezekiel (1930) formula:

$$R^2_{\text{adj}} = 1 - (1 - R^2_{\mathbf{Y}|\mathbf{X}})\frac{(n-1)}{(n-m-1)} \qquad \text{eqn 5}$$

where m is the number of explanatory variables in **X** or, more precisely, the rank of the variance-covariance matrix of **X**.

- The F-statistic for the overall test of significance is constructed as follows (Miller 1975):

$$F = \frac{R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}} \Big/ mp}{(1 - R^2_{\mathbf{Y}_{\text{stand}}|\mathbf{X}}) \Big/ (n-m-1)p} \qquad \text{eqn 6}$$

This statistic is used to perform the overall test of significance of the canonical relationship. The null hypothesis of the test is H$_0$: the strength of the linear relationship, measured by the canonical R^2, is not larger than that which would be obtained for unrelated **Y** and **X** matrices of the same sizes [Note: in the absence of relationship, the expected value of R^2 is not 0 but $m/(n\text{–}1)$].

When the variables of **Y** are standardized, the F-statistic (eqn 6) can be tested for significance using the Fisher–Snedecor F-distribution with d.f.$_1$ = mp and d.f.$_2$ = $p(n - m - 1)$; p is the number of response variables in **Y**; m parameters were estimated for each of the p multiple regressions used to compute the vectors of fitted values forming the p columns of $\hat{\mathbf{Y}}$; hence, a total of mp parameters were estimated. This is why there are mp degrees of freedom attached to the numerator of F (d.f.$_1$). Each multiple regression equation has degrees of freedom equal to $(n - m - 1)$, so the number of degrees of freedom of the denominator, d.f.$_2$, is p times $(n - m - 1)$. Miller (1975) conducted numerical simulations in the multivariate normal case, with combinations of m and p from 2 to 15 and sample sizes of 30 to 160. He showed that eqn 6 produced distributions of F values that were very close to theoretical F-distributions with the same numbers of degrees of freedom. Additional simulations that we conducted (reported in Appendix S1) confirmed that this parametric test of significance had correct levels of type I error when **Y** was standardized. This was not the case, however, for nonstandardized response variables. In our simulations, the columns of **Y** were random numbers drawn from linearly independent statistical populations. Further simulations should be performed to check the validity of Miller's parametric test when there are correlations among the columns of **Y**.

In many analyses, the response variables should not be standardized prior to RDA. With community composition data in ecology (species abundance data), for instance, the variances of the species should be preserved in most analyses because abundant and rare species do not play the same roles in ecosystems. Our simulation results (Appendix S1) show that parametric tests should not be used when the **Y** variables have unequal variances, especially when the error is not normal. Permutation tests always had correct levels of type I error in our results. For permutation tests, one can simplify the equation of the F-statistic and eliminate the constant p:

$$F = \frac{R^2_{\mathbf{Y}|\mathbf{X}} \Big/ m}{(1 - R^2_{\mathbf{Y}|\mathbf{X}}) \Big/ (n-m-1)} \qquad \text{eqn 7}$$

This simplification does not change the value of F. Equation 7 is the one used in programs of canonical analysis, such as CANOCO and vegan's *rda()*, designed to analyse empirical user's data. The F-statistic is tested by permutation in these programs.

STATISTICS, PARTIAL RDA

- For analysis in the presence of **W** containing q covariables (partial RDA), the F-statistic is constructed as follows (ter Braak & Šmilauer 2002):

$$F = \frac{\text{SS}(\mathbf{Y}_{\text{fit}})/m}{\text{SS}(\mathbf{Y}_{\text{res}})/(n-m-q-1)} \qquad \text{eqn 8}$$

There are several ways of computing the sum-of-squares of the fitted values SS($\mathbf{Y}_{\text{fit}}$) and residuals SS($\mathbf{Y}_{\text{res}}$) in the partial RDA case. The most convenient is the following: $\text{SS}(\mathbf{Y}_{\text{fit}}) = \text{SS}(\mathbf{Y}_{\text{fit}|(\mathbf{X}+\mathbf{W})}) - \text{SS}(\mathbf{Y}_{\text{fit}|\mathbf{W}})$ and $\text{SS}(\mathbf{Y}_{\text{res}}) = \text{SS}(\mathbf{Y}) - \text{SS}(\mathbf{Y}_{\text{fit}|(\mathbf{X}+\mathbf{W})})$. (**X** + **W**) designates the concatenation of **X** and **W** in a single matrix. $\mathbf{Y}_{\text{fit}}$ was noted $\hat{\mathbf{Y}}$ in eqn 1, which did not involve covariables **W**.

- The semipartial R^2, $R^2_{\mathbf{Y}|\mathbf{X}_{\text{res}|\mathbf{W}}}$, is the proportion of explained variation with respect to the total variation in **Y**. This is the most widely used R^2 statistic in partial RDA. It is the R^2 of the simple RDA of **Y** by $\mathbf{X}_{\text{res}|\mathbf{W}}$:

$$R^2_{\mathbf{Y}|\mathbf{X}_{\text{res}|\mathbf{W}}} = \text{SS}(\mathbf{Y}_{\text{fit}})/\mathbf{SS}(\mathbf{Y}) \qquad \text{eqn 9}$$

- The partial R^2, $R^2_{\mathbf{Y}_{\text{res}|\mathbf{W}}|\mathbf{X}_{\text{res}|\mathbf{W}}}$, is the proportion of explained variation with respect to the total variation in **Y** residualized on the matrix of covariables **W**. Although more rarely used, it can be computed as the R^2 of the simple RDA of $\mathbf{Y}_{\text{res}|\mathbf{W}}$ by $\mathbf{X}_{\text{res}|\mathbf{W}}$:

$$R^2_{\mathbf{Y}_{\text{res}|\mathbf{W}}|\mathbf{X}_{\text{res}|\mathbf{W}}} = \text{SS}(\mathbf{Y}_{\text{fit}})/\text{SS}(\mathbf{Y}_{\text{res}|\mathbf{w}}) \qquad \text{eqn 10}$$

Methods to test the significance of individual canonical axes in RDA

The null hypothesis for the test of significance of the *j*th axis is H_0: the linear dependence of the response variables **Y** on the explanatory variables **X** is less than *j*-dimensional. More informally, the null hypothesis is that the *j*th axis under test explains no more variation than a random axis of the same order (*j*), for matrices **Y** and **X** (in the presence of covariables **W**, if applicable), given the variation explained by the previously tested axes.

TEST ALL INDIVIDUAL CANONICAL AXES SIMULTANEOUSLY

The first and simplest method can be described as follows (covariables **W** are not considered in this first form of test):

1. Compute the RDA of **Y** by **X**. Extract all canonical eigenvalues.

2. A rough test could be based on the eigenvalues themselves. For a better test design, for each canonical eigenvalue λ_j, we will compute *F*-statistics using the following formulas:

$$F1_j = \lambda_j/((\text{SS}(\mathbf{Y}) - \sum_{i=1}^{j} \lambda_i)/(n-1-m)) \qquad \text{eqn 11}$$

$$F2_j = \lambda_j/((\text{SS}(\mathbf{Y}) - \sum_{i=1}^{k} \lambda_i)/(n-1-m)) \qquad \text{eqn 12}$$

where *n* is the number of objects, *m* is the rank of the variance-covariance matrix of **X**, *k* is the number of canonical eigenvalues and SS(**Y**) is the total sum-of-squares in **Y**. The first formula for the *F*-statistic is related to the formula found in the CANOCO manual (ter Braak & Šmilauer 2002, p. 51) to test the first canonical eigenvalue; it uses the sum of canonical eigenvalues up to and including the λ_j being tested. The second formula uses the sum of *all k* canonical eigenvalues in the denominator of *F* in the same way as the marginal method below. In both formulas, the eigenvalue λ_j in the numerator could be divided by *m*. That constant, as well as the number of degrees of freedom $(n - 1 - m)$ in the denominator, has no influence on the results of permutation tests.

3. Permute the rows of **Y** at random, obtaining matrix **Y***.

4. Compute the canonical analysis of **Y*** by **X**. Obtain all canonical eigenvalues. For each canonical eigenvalue of the permuted analysis, compute the $F1_j$* and $F2_j$* statistics under permutation, using the formulas above.

5. Repeat steps 3 and 4 a large number of times, say 999 times, to obtain estimates of the distributions of $F1_j$* and $F2_j$* under permutation. Add the reference values of $F1_j$ and $F2_j$ to their respective distributions (Hope 1968).

6. Calculate the associated probabilities as the number of cases in the distributions that are larger than or equal to the reference values, divided by the number of permutations plus 1.

That method is appealing because of its simplicity. Computation is faster than with the two following methods. Because it is the simplest one, it may be appealing to the developers of new programs; this is why it is described, tested and discussed here. Our simulation results will show that this method should not be used.

MARGINAL AND FORWARD TEST OF THE CANONICAL AXES

The marginal and forward tests of canonical axes use *F*-statistics similar to those described in eqns 11 and 12, respectively. They differ from the simultaneous test described in section 'Test all individual canonical axes simultaneously' except for the first canonical axis. For the second and later canonical axes, the forward and marginal tests are based on a partial RDA, even in the case where there were initially no covariables. For testing the *j*th axis ($j > 1$), the lower-numbered canonical axes (1, 2 … (*j*–1)) are added to the matrix of covariables **W**. Calculation of the residuals, which are permuted and added to the unpermuted fitted values during the permutation test for each axis, is performed on the covariables **W** incremented with the previously tested axes. The test cannot be performed simultaneously for all axes because the residual sum-of-squares, which is computed using the covariables **W** incremented with the previously tested axes, is one of the elements that form the denominator of the *F*-statistic and it varies for each axis. Furthermore, for partial regression and partial canonical analysis, Anderson & Legendre (1999) have shown that permutation of the residuals of either the reduced or the full model (see 'Permutation methods' below) is preferable to permutation of the raw data.

It is very difficult to describe the methods for testing the canonical axes in any detail using sentences and paragraphs. Considering the popularity of the R statistical language which is used by many ecologists and other application scientists, we append to the paper (Data S1) two R functions, called *marginal.test()* and *forward.test()*, which serve as complete descriptions of the marginal and forward methods; to keep these functions simple, the analysis is performed without true covariables and the functions are written without shortcuts or optimizations. Detailed comments have been added to the code to make it understandable even by those who are not fluent in R. The purpose of these functions is to unambiguously describe the two methods. More complex functions involving covariables are presented in Data S2; they are called *test.axes.canoco()* and *test.axes.cov()*. These four functions do not call compiled code that would increase computation efficiency during permutation testing. They are not intended for routine testing of canonical eigenvalues, although they do produce correct, publishable results.

Marginal test

The marginal method, which is based on the approach used in marginal tests of significance in partial regression, is computed as follows. A matrix of covariables may be present in the analysis; it is included in all steps of the description.

1. Compute the RDA of **Y** by **X** in the presence of covariables **W**, if any; see 'Partial RDA' in 'Background: the algebra of redundancy analysis'. Extract the eigenvalues as well as the canonical axes that are linear combinations of the explanatory variables **X** (i.e. the object score ordination axes).

2. Test the significance of the successive canonical eigenvalues, λ_j, using the following *F*-statistic:

$$\begin{aligned} F3_j &= \lambda_j/(\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|(\mathbf{X}+\mathbf{W})})/(n-1-m)) \\ &= \lambda_j(n-1-m)/\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|(\mathbf{X}+\mathbf{W})}) \qquad \text{eqn 13} \end{aligned}$$

In the marginal method, the denominator of the *F*-statistic, $\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|(\mathbf{X}\ +\ \mathbf{W})})$, is always (i.e. for the tests of all axes) the residual sum-of-squares of the model including all explanatory variables **X** and all covariables **W**, if any. Permutation of the raw data can be used to test the first canonical eigenvalue if there is no matrix **W** in the analysis. Permutation of the raw data and permutation of the residuals of the reduced model are identical in tests without covariables (Legendre & Legendre 1998, Table 11·7). If there are covariables in the analysis, and for the test of all successive canonical axes, use permutation of the residuals of the reduced or full model (see 'Permutation models' below). In the *marginal.test()* function in Data S1, permutation of the residuals of the reduced model is used to test the canonical axes.

Forward test

A forward approach to the test of the canonical axes has been implemented in the CANOCO program since version 3.10 (ter Braak 1990). It can be described as follows. The following description includes a matrix of covariables that may be present in the analysis.

1. Compute the RDA of **Y** by **X** in the presence of covariables **W**, if any; see 'Partial RDA' above. Set aside the vector of canonical eigenvalues and the matrix containing the canonical axes that are linear combinations of the explanatory variables **X**.

2. Test the significance of the successive canonical eigenvalues, λ_j, using the following *F*-statistic:

$$\begin{aligned} F4_j &= \lambda_j/((\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|[\mathbf{W}+\mathrm{Axis}(1)\ \mathrm{to}\ \mathrm{Axis}(j-1)]}) - \lambda_j)/(n-1-m)) \\ &= \lambda_j(n-1-m)/(\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|[\mathbf{W}+\mathrm{Axis}(1)\ \mathrm{to}\ \mathrm{Axis}(j-1)]}) - \lambda_j) \qquad \text{eqn 14} \end{aligned}$$

where λ_j is the eigenvalue under test. λ_j is the first eigenvalue from the analysis of **Y** [or permuted **Y** during the permutations] by **X** in the presence of covariables **W** and the previously tested axes 1 to (*j*–1), which are added as columns to **W**. $\mathrm{SS}(\mathbf{Y}_{\mathrm{res}|[\mathbf{W}\ +\ \mathrm{Axis}\ (1)\ \mathrm{to}\ \mathrm{Axis}\ (j-1)]})$ is the residual sum-of-squares of that partial RDA model. From that quantity, we subtract eigenvalue λ_j, which is recomputed during each permutation. Permutation of the raw data can be used to test the first canonical eigenvalue if there is no matrix **W** in the analysis. If there is, and for the test of all successive canonical axes, use permutation of the residuals of the reduced or full model (see 'Permutation models' below). In the *forward.test()* function in Data S1, permutation of the residuals of the reduced model is used to test the canonical axes; see the section 'Permutation methods' below.

PARAMETRIC TEST

Besides the three methods described above, Lazraq & Cléroux (2002) proposed a parametric method to test the significance of the successive components in RDA under the assumption of multinormality. Takane & Hwang (2005) showed, however, that the test of Lazraq and Cléroux lead to strongly biased results. Takane & Hwang (2005) also suggested that the permutation test used by Takane & Hwang (2002) in generalized canonical correlation analysis can be adapted to RDA when the multivariate normality assumption does not hold; it leads to the forward permutation procedure described in the present paper. In support of their suggestion, Takane & Hwang (2005) cited some of the simulation results reported in the present paper, which they had been shown during a seminar by PL in 2005.

PERMUTATION METHODS

In the marginal and forward methods, significance tests of the canonical axes involve either unrestricted permutation of the residuals of the reduced model, a method proposed by Freedman & Lane (1983), or permutation of the residuals of the full model, a method proposed by ter Braak (1990, 1992). These methods are described in Anderson & Legendre (1999) for multiple regression and in Legendre & Legendre (1998, section 11·3) for RDA and CCA. Permutation of the raw data is another possible option; it will be compared to the permutation of the residuals of the reduced model in the simulations reported in section 'Results and Discussion'.

- In permutation of the raw data (method = 'direct' in vegan and in our simulation software), the rows of **Y** are permuted at random to produce the matrix of permuted response data **Y***.
- In permutation of the residuals of the reduced model, one computes the matrix of fitted values $\mathbf{Y}_{\mathrm{fit}|\mathbf{W}}$ and the matrix of residuals $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$ of the multivariate regression of **Y** on the matrix of covariables **W**. The rows of $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$ are permuted, producing matrix $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$*. The matrix of permuted response data, **Y***, is obtained by adding $\mathbf{Y}_{\mathrm{fit}|\mathbf{W}}$ (unpermuted) to $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$*.
- In permutation of the residuals of the full model, one computes the matrix of fitted values $\mathbf{Y}_{\mathrm{fit}|(\mathbf{X}\ +\ \mathbf{W})}$ and the matrix of residuals $\mathbf{Y}_{\mathrm{res}|(\mathbf{X}\ +\ \mathbf{W})}$ of the multivariate regression of **Y** on the matrix obtained by concatenation of **X** and **W** by columns into a single matrix. The rows of $\mathbf{Y}_{\mathrm{res}|(\mathbf{X}\ +\ \mathbf{W})}$ are permuted, producing matrix $\mathbf{Y}_{\mathrm{res}|(\mathbf{X}\ +\ \mathbf{W})}$*. The matrix of permuted response data, **Y***, is obtained by adding $\mathbf{Y}_{\mathrm{fit}|(\mathbf{X}\ +\ \mathbf{W})}$ (unpermuted) to $\mathbf{Y}_{\mathrm{res}|(\mathbf{X}\ +\ \mathbf{W})}$*.

In version 3.x of the CANOCO program, the default method for the test of significance of the first canonical eigenvalue was permutation of the residuals of the reduced model, and permutation of the residuals of the full model for the overall test of significance of the canonical relationship. In version 4.x of CANOCO, the default became permutation of the residuals of the reduced model in all cases. The default method for the marginal test in vegan was permutation of the raw data (method = 'direct') until version 1.14; the default was changed to method = 'reduced' in version 1.15.

Permutation of the residuals of the reduced and the full model were found by Anderson & Legendre (1999) to produce equivalent results. Only permutation of the residuals of the reduced model was used in the simulations reported in the present paper.

Besides these methods, one can also permute **Y** in a way imposed by the logic of a problem. The most important methods of restricted permutation are permutation within the levels of a factor or block which is used as a covariable in the study and loop permutation along a time series or toroidal permutation of the points on a geographical surface.

Software

As mentioned above, Data S1 of this paper contains functions that describe the marginal and forward methods for simple RDA, using R code. Data S2 describes two R functions designed for testing canonical axes in simple or partial RDA. These two functions are programmed differently, following the two approaches for partial RDA described in the 'Partial RDA' subsection of 'Background: the algebra of redundancy analysis' of this paper. They produce identical results. For users who are analysing real data, the R package 'vegan' (Oksanen *et al.* 2010) offers canonical analysis by RDA and CCA with covariables **W**, with tests of significance of the canonical axes through the marginal test. The marginal method is implemented in the *permutest.cca()* function, which carries out the tests of significance of the canonical axes when users call the *anova.cca()* function after canonical analysis by the functions *rda()* and *cca()*. The program CANOCO (ter Braak & Šmilauer 2002) offers tests of significance of the canonical axes through the forward method.

Simulation method

Simulations were carried out to compare the statistical properties of three methods developed to test the significance of the canonical axes in RDA described in sections 'Test all individual canonical axes simultaneously' and 'Marginal and forward test of the canonical axes'. The simulations were designed specifically for that purpose, by opposition with simulations that could be performed to illustrate the use of RDA and its tests of significance in a wide range of real data analyses. One of the limitations of any simulation study is that not all possible cases can be covered. So, in this paper, we limited our simulations to key situations where a known number of canonical axes were generated. The three methods will be compared as to their ability to detect the correct number of significant axes in the simulated data. A method that detects significant axes not corresponding to linear relationships built into the data will be declared incorrect. In practice, then, we generated data having a known number of linear relationships (canonical dimensions) to determine whether the testing procedures found the correct number of dimensions. This is the role of the blocks of variables created in the response matrix **Y** and described below. This data structure is a simplification, but not an oversimplification, of the kind of relationships that would be found in real data, for example when analysing the relationships between species and environmental variables.

Type I error consists in finding 'false positives', or false significant results, during statistical tests when there is no effect – here, a linear relationship between the response and explanatory matrices. To estimate the rate of type I error in the tests of the canonical axes, pairs of **Y** and **X** matrices were generated using random deviates and tested for significance. A test of significance is valid if the probability of type I error is no greater than the significance level α, for any α value (Edgington 1995, p. 37). We used the following values to generate the data for the simulations to study type I error: n = 20 or 100; p (number of variables in **Y**) = 3, 5, or 8; m (number of variables in **X**) = 3, 5, or 8. Error was normally distributed.

Power is the ability of a statistical method (here the test of individual canonical axes in RDA) to detect a relationship when one is present in the data. The difficulty in the present study was to generate data that contained a known number of linear canonical relationships, that is, a known number of dimensions, to check that the testing procedures for the canonical axes found the correct number of dimensions. That is the role of the blocks of variables generated in the response matrix **Y**. The generation method is described in the following paragraphs.

To give a simple example, we could have generated a single variable in **Y** related to a single variable in **X** as follows:

$$\mathbf{Y}(i,j) = \mathbf{X}(i,j) + 0{\cdot}2\varepsilon \quad \text{where } \varepsilon \sim \mathrm{N}(0,1) \qquad \text{eqn 15}$$

Analysis of that pair would have been expected to produce a single significant canonical eigenvalue.

In our simulations, we generated data that contained 1–4 blocks, each containing 1–3 variables in **Y**; each block was related to two variables in **X**. Because of that, when there was more than a single **y** variable per block, more canonical axes were created than the number of significant axes, which was equal to the number of blocks of **Y** variables. For example, create four independent random normal variables following N(0,1), called $\mathbf{x}_1 \ldots \mathbf{x}_4$, and six more random normal variables following N(0,1), called $\boldsymbol{\varepsilon}_1 \ldots \boldsymbol{\varepsilon}_6$. Then, create two blocks in **Y**, each block containing three **y** variables:

$$\text{block 1: } \mathbf{y}_1 = 0{\cdot}5\mathbf{x}_1 + 0{\cdot}5\mathbf{x}_2 + c\varepsilon_1, \mathbf{y}_2 = 0{\cdot}5\mathbf{x}_1 + 0{\cdot}5\mathbf{x}_2 + c\varepsilon_2,$$
$$\mathbf{y}_3 = 0{\cdot}5\mathbf{x}_1 + 0{\cdot}5\mathbf{x}_2 + c\varepsilon_3, \text{ and}$$
$$\text{block 2: } \mathbf{y}_4 = 0{\cdot}5\mathbf{x}_3 + 0{\cdot}5\mathbf{x}_4 + c\varepsilon_4, \mathbf{y}_5 = 0{\cdot}5\mathbf{x}_3 + 0{\cdot}5\mathbf{x}_4 + c\varepsilon_5,$$
$$\mathbf{y}_6 = 0{\cdot}5\mathbf{x}_3 + 0{\cdot}5\mathbf{x}_4 + c\varepsilon_6.$$

In this example, the coefficient c determines the importance of the random error component; that coefficient will vary in the simulations reported in the next section. The **Y** variables within each block are all related to a linear model of the same pair of **X** variables but differ from one another by their random component $\boldsymbol{\varepsilon}$, which differs from variable to variable. So, an analysis of that pair of matrices **Y** and **X** would be expected to produce two significant canonical axes.

We used the following values for the simulations for power reported in this paper: No. blocks = 1–4, No. **y** per block = 1–3, n = 20 or 100. As two **X** variables were generated for each block in **Y**, there are m = 2–8 variables in **X** and p = 1–12 variables in **Y**. Error distributions: normal, exponential, cubed exponential, as in Manly (1997) and Anderson & Legendre (1999). The weight of the error component in the generation of the **y** variables was c = 0·2, 0·5, or 0·8.

In both the type I error and the power studies, each simulation run consisted of the analysis of 1000 pairs of independently generated data matrices. The permutation tests involved 999 random permutations. The significance level of the tests was α = 0·05.

Results and Discussion

The simulation results lead to the following observations.

SIMULTANEOUS TEST OF ALL CANONICAL EIGENVALUES

The simulation results reported in the first six rows of Table B1 (Appendix S2), parts a and b, show that in the absence of a relationship between **Y** and **X**, the rejection rates of H_0 were always close to the nominal significance level, here 5%. The tests had correct rates of type I error in that situation for both ways of computing the F-statistic ($F1$ and $F2$, eqns 2 and 3). When relationships were present in the data (*Power and type I error* sections, Table B1 in Appendix S2), the tests had good power to detect the significant axes displaying the relationships; the expected number of significant axes is the number of blocks of **Y** variables ('No. blocks' column). The tests of the following eigenvalues, which did not display relationships built in the data, should, however, have had rejection rates close to the significance level (α = 0·05) or lower. The results show that the rejection rates for these axes had highly inflated levels of type I error (values in bold in the table). So that form of test is invalid. As a consequence, the simultaneous test of all canonical eigenvalues should not be used.

MARGINAL AND FORWARD METHODS, NORMAL ERROR

The results presented in Fig. 1 and Tables B2 and B3 in Appendix S2 indicate that these two methods have very similar properties: in the absence of a relationship between **Y** and **X**, the rejection rates of H_0 were always close to or lower than the significance level α = 0·05 (Tables B2 and B3 in Appendix S2, 'Type I error' sections). When relationships were present in the data, the tests had good power to detect the axes displaying the relationships (Tables B2 and B3 in Appendix S2, 'Power and type I error' sections); the expected number of significant axes is the number of blocks of **Y** variables ('No. blocks' column). For all axes that were not expected to display a significant relationship, the rejection rates (type I error) were always lower than α = 0·05. These two forms of test are thus valid, even though their conservative behaviour for the nonsignificant axes indicates a slight loss of power for the significant axes after the first one.

COMPARISON OF THE 'DIRECT' AND 'REDUCED' PERMUTATION METHODS

Does the choice of a permutation method, 'direct' or 'reduced', make a difference for the marginal method? Simulations were conducted with increased weights for the error component in the generation of the **Y** variables (parameter c = 0·5 and 0·8, instead of 0·2 in Tables B2 and B3, Appendix S2), in the hope of displaying a difference of power between the two permutation methods. The results presented in Table B4 in Appendix S2 for the marginal test show that permutation of the residuals of the reduced model has more power than permutation of the raw data to detect a relationship when there is a large amount of error in the data, especially when n is small. For c = 0·8 for example, compare rows of results with the same values of n and 'No. blocks': the rejection rates for the axes that were expected to be significant, following the first one, are always higher for method = 'reduced' than for method = 'direct'. As expected, power is much higher for n = 100 than for n = 20.

INFLUENCE OF INCREASING WEIGHT OF THE ERROR COMPONENT

As expected, increasing the weight of the error component in the generation of **Y** (parameter c) reduced the power of the marginal and forward tests of the canonical axes (Tables B2 and B3, Appendix S2: c = 0·2; Tables B4 and B5: c = 0·5 and 0·8; Fig. 1).

INFLUENCE OF ERROR TYPE

For reduced model permutations, the results for normal and exponential error are similar (Fig. 1). For cubed exponential error (very highly skewed data), the power to detect significant relationships almost completely vanishes because the linear component of the **X**–**Y** relationships is weak. Detailed results are presented in Tables B6 and B7 in Appendix S2. For highly skewed data, such as those generated here using cubed exponential deviates as the error term, one might more easily identify relationships between **y** and **X** using De'ath's (2002) multivariate regression tree, which aims at identifying breaks in the response data values that correspond to thresholds in the explanatory variables, instead of linear relationships with the explanatory variables as it is the case in RDA.

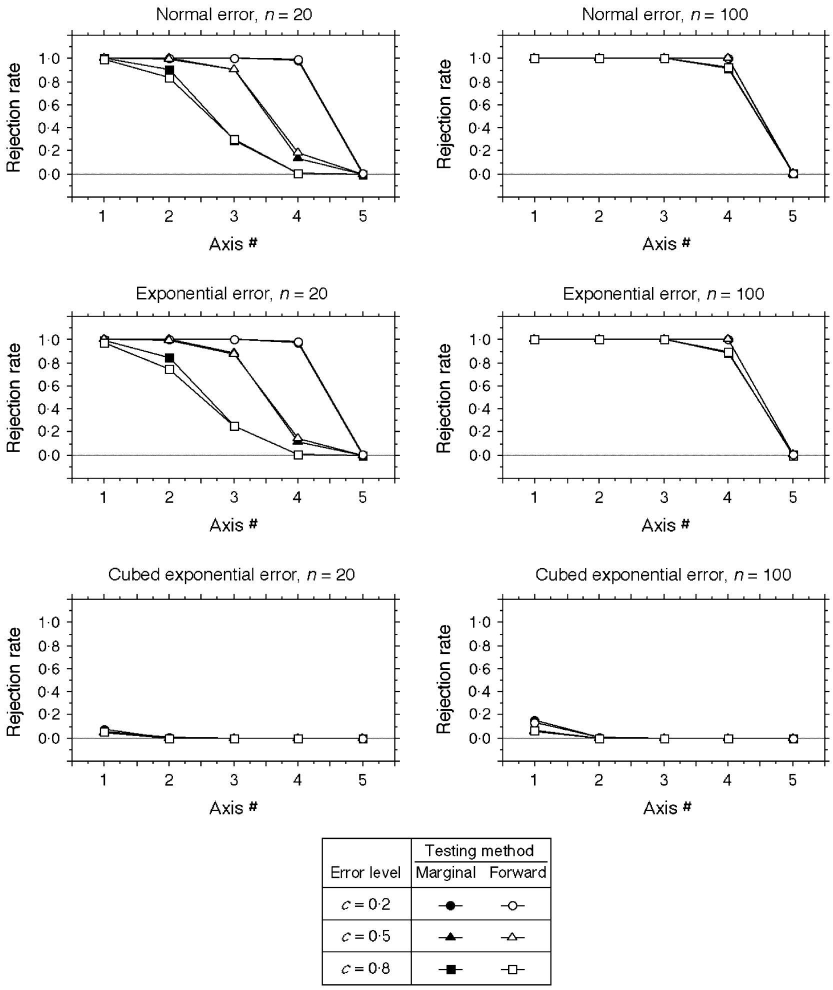

Fig. 1. Comparison of the marginal and forward testing methods in tests of canonical axes #1–5 for $n = \{20, 100\}$ and different types of error, reduced-model permutations. There were four blocks of **y** variables in these simulations, i.e. four distinct linear relationships between **Y** and **X**, $p = 12$ variables in **Y**, and $m = 8$ variables in **X**. The black symbols (marginal method) are often hidden behind the white symbols (forward method). The data are from Tables B2 to B7 (Appendix S2).

There is no need for an adjustment for multiple testing when testing the canonical axes. An overall test of the canonical relationship (canonical R^2) must be carried out, ideally before the canonical axes are computed, and certainly before the axes are tested and examined. So one is not interested here in adjusting the significance level of individual tests to obtain a fixed (e.g. 5%) experimentwise error rate, but only in a test of significance that has a correct rate of type I error for each axis.

Statistical significance is of course not the same as biological importance. If an axis is not statistically significant, it usually does not warrant mapping, biplotting or biological interpretation, with perhaps an exception: when the number of sites is very small and power is low, one might still want to draw a biplot and examine some of the first nonsignificant axes. On the other hand, even when an axis is statistically significant, it may not be worth interpreting when it explains little variation; this may happen when the number of objects is large. We advise that the results of tests of statistical significance should not be used blindly.

Conclusion

In this paper, we addressed the following questions through numerical simulations:

- Which of the three testing methods produced correct type I error for the test of the successive canonical eigenvalues? The computationally more simple method which consists of testing all axes with a single set of permutations produced incorrect type I error for all axes except the first one; so it is invalid and should not be used to test the significance of canonical axes in RDA. The marginal and forward procedures are both valid as they displayed type I error rates no greater than the significance level α; we used $\alpha = 0{\cdot}05$ in this simulation study. These two permutation methods produced comparable results and were equally powerful for the range of conditions studied in our simulations (Tables B1–B3 in Appendix S2).
- How does permutation of the raw data compare to permutation of the residuals in the marginal and forward tests? Permutation of the residuals of the reduced model provided greater power than permutation of the raw data in the tests of

the axes following the first one (Tables B4 and B5 in Appendix S2).

- What is the effect of the type of error on the power of the marginal and forward tests? The importance of the loss of power depended on the number of observations *n* and on the type of error: with normal or exponential error, the loss was very slight when using permutation of the residuals [Tables B4 (method = 'reduced') and B5–B7 in Appendix S2].

The R functions found in Data S1 provide the first formal description of the 'marginal' and 'forward' testing methods.

Acknowledgements

We are grateful to Daniel Borcard, Stéphane Dray, as well as two anonymous reviewers for discussion of the ideas presented in the Introduction of this paper and for useful suggestions about the manuscript, and to Guillaume Guénard for assistance during the simulation work. This research was supported by NSERC grant no. 7738-07 to P. Legendre.

References

Anderson, M.J. & Legendre, P. (1999) An empirical comparison of permutation methods for tests of partial regression coefficients in a linear model. *Journal of Statistical Computation and Simulation*, **62**, 271–303.

Blanchet, F.G., Legendre, P. & Borcard, D. (2008) Modelling directional spatial processes in ecological data. *Ecological Modelling*, **215**, 325–336.

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.

ter Braak, C.J.F. (1986) Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology*, **67**, 1167–1179.

ter Braak, C.J.F. (1987) The analysis of vegetation-environment relationships by canonical correspondence analysis. *Vegetatio*, **69**, 69–77.

ter Braak, C.J.F. (1990) *Update Notes: CANOCO Version 3.10*. Agricultural Mathematics Group, Wageningen. http://www.biometris.wur.nl/UK/Staff/Cajo+ter+Braak/Publications/.

ter Braak, C.J.F. (1992) Permutation versus bootstrap significance tests in multiple regression and ANOVA. *Bootstrapping and Related Techniques* (eds K.-H. Jöckel, G. Rothe & W. Sendler), pp. 79–86. Springer-Verlag, Berlin.

ter Braak, C.J.F. (1995). Ordination. *Data Analysis in Community and Landscape Ecology* (eds R.H.G. Jongman, C.J.F. ter Braak & O.F.R. van Tongeren), pp. 91–173. Cambridge University Press, Cambridge.

ter Braak, C.J.F. & Prentice, I.C. (1988) A theory of gradient analysis. *Advances in Ecological Research*, **18**, 271–317.

ter Braak, C.J.F. & Šmilauer, P. (2002) *CANOCO Reference Manual and CanoDraw for Windows user's Guide – Software for Canonical Community Ordination (Version 4.5)*. Microcomputer Power, Ithaca, New York.

De'ath, G. (2002) Multivariate regression trees: a new technique for constrained classification analysis. *Ecology*, **83**, 1105–1117.

Dray, S., Legendre, P. & Peres-Neto, P. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbor matrices (PCNM). *Ecological Modelling*, **196**, 483–493.

Edgington, E.S. (1995) *Randomization Tests*, 3rd edn. Marcel Dekker, Inc., New York.

Ezekiel, M. (1930) *Methods of Correlation Analysis*. John Wiley and Sons, New York.

Freedman, D. & Lane, D. (1983) A nonstochastic interpretation of reported significance levels. *Journal of Business and Economic Statistics*, **1**, 292–298.

Griffith, D. (2000) A linear regression solution to the spatial autocorrelation problem. *Journal of Geographical Systems*, **2**, 141–156.

Hope, A.C.A. (1968) A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society Series B*, **30**, 582–598.

Lazraq, A. & Cléroux, R. (2002) Testing the significance of the successive components in redundancy analysis. *Psychometrika*, **67**, 411–419.

Legendre, P. & Legendre, L. (1998) *Numerical Ecology*, 2nd English edn. Elsevier Science BV, Amsterdam.

Manly, B.J.F. (1997) *Randomization, Bootstrap and Monte Carlo Methods in Biology*, 2nd edn. Chapman and Hall, London.

Miller, J.K. (1975) The sampling distribution and a test for the significance of the bimultivariate redundancy statistic: a Monte Carlo study. *Multivariate Behavioral Research*, **10**, 233–244.

Miller, J.K. & Farr, S.D. (1971) Bimultivariate redundancy: a comprehensive measure of interbattery relationship. *Multivariate Behavioral Research*, **6**, 313–324.

Oksanen, J., Blanchet, F.G., Kindt, R., Legendre, P., O'Hara, R.B., Simpson, G.L., Solymos, P., Stevens, M.H.H. & Wagner, H. (2010) *Vegan: Community Ecology Package*. R package version 1.17-4. http://cran.r-project.org/web/packages/vegan.

Rao, C.R. (1964) The use and interpretation of principal component analysis in applied research. *Sankhyā: The Indian Journal of Statistics, Series A*, **26**, 329–358.

Takane, Y. & Hwang, H. (2002) Generalized constrained canonical correlation analysis. *Multivariate Behavioral Research*, **37**, 163–195.

Takane, Y. & Hwang, H. (2005) On a test of dimensionality in redundancy analysis. *Psychometrika*, **70**, 271–281.

Tso, M.K.-S. (1981) Reduced-rank regression and canonical analysis. *Journal of the Royal Statistical Society Series B*, **43**, 183–189.

Received 14 May 2010; accepted 19 October 2010
Handling Editor: Emmanuel Paradis

Supporting Information

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

Appendix S1. Simulation results, test of the global RDA R-square.

Appendix S2. Simulation results, test of individual canonical axes.

Data S1. Description of the marginal and forward methods for testing the canonical axes in simple redundancy analysis (RDA) without covariables. The methods are described with R functions.

Data S2. R functions for testing the canonical axes in partial RDA.

Data S3. Simulation method, test of the global RDA R-square.

Data S4. Simulation methods, test of individual canonical axes.

A zipped folder 'Legendre_et al._R_functions.zip' contains R-language functions for testing canonical axes, without and in the presence of a matrix of covariables (subfolder 'R functions for testing axes'), as well as the R functions used during the simulations reported in the paper (simulation methods in Data S3 and S4, results in Appendices S1 and S2).

As a service to our authors and readers, this journal provides supporting information supplied by the authors. Such materials may be re-organized for online delivery, but are not copy-edited or typeset. Technical support issues arising from supporting information (other than missing files) should be addressed to the authors.

Appendices to:

Legendre, P., Oksanen, J. & ter Braak, C. J. F. (2011) Testing the significance of canonical axes in redundancy analysis. *Methods in Ecology & Evolution*, **2**, 269–277.

APPENDIX S1

SIMULATIONS RESULTS, *F*-TEST OF THE GLOBAL RDA R-SQUARE

This Appendix reports results of the simulations conducted to test the type I error of the global *F*-test of significance in RDA under different conditions. Type I error occurs in the global test of the canonical R^2 when a test result is significant for matrices **Y** and **X** between which there is no linear relationship. A test of significance is valid if the probability of type I error is no greater than the significance level α, for any α value (Edgington 1995, p. 37); α was 0.05 in our simulations.

In a first series of simulations, we verified the finding of Miller (1975) that the *F*-statistic computed as in eq. 6 produces valid tests of significance for standardized data. Data were generated independently in matrices **Y** and **X**, using the software described in document Data S3; this insured that there were no linear relationships between the variables of **Y** and **X** except by chance. The columns of **Y** were random numbers drawn from linearly independent statistical populations. The error in **Y** was either normal, exponential, or cubed exponential (exponential deviates to the exponent 3, which provided extremely non-normal error), following previous simulations studies of the same type (e.g. Manly 1997, Anderson and Legendre 1999). The variables in **Y** were standardized prior to testing. The results (Table A1) show that the parametric test had correct levels of type I error for all combinations of the simulation parameters. The permutation test also had correct levels of type I error.

Then, we generated random variables in **Y** with equal population variances, but we did not standardized matrix **Y** prior to the tests. The results (Table A2) show that while the test of the canonical relationship remained valid for normal and exponential error, the rejection rates were inflated (i.e. higher than the significance level) for extremely non-normal error. The permutation test had correct levels of type I error in all simulations.

Finally, we generated random variables in **Y** with unequal population variances. The generated random deviates were multiplied by values chosen at random, in each simulation independently, from a random uniform distribution between 1 and 10. There was inflation of the rejection rate of the parametric test of significance for all types of error distributions. The effect was less important for random normal error and more important for exponential and cubed exponential deviates. The permutation test had correct levels of type I error in all simulations.

We conclude that the parametric method of testing the canonical relationship proposed by Miller (1975) can be safely used only when the variables in **Y** are standardized prior to canonical analysis. It can also be used for non-standardized variables **Y** drawn from normally distributed variables with equal population variances. In all other cases, permutation tests should be used. The permutation tests can actually be used safely in all cases, including those where the Miller parametric test can be used.

Table A1. Simulation results for type I error in *F*-tests of significance of the global R^2 statistic for standardized **Y** variables: parametric test (Miller *F*-statistic, eq. 6) and permutation test rejection rates of H_0 at the 5% significance level over 1000 simulations. Permutation tests involved 999 random permutations. *n*: number of observations; *p*: number of response variables in **Y**; *m*: number of explanatory variables in **X**. **Y** and **X** were generated independently in these simulations.

n	*p*	*m*	Rejection rates	
			Parametric	Permutation
a) Normal error				
10	1	1	0.047	0.051
20	1	1	0.047	0.047
20	5	5	0.039	0.051
20	10	5	0.040	0.049
20	20	10	0.029	0.039
50	5	5	0.038	0.041
50	1	1	0.055	0.055
50	10	5	0.052	0.050
50	20	10	0.043	0.040
100	1	1	0.058	0.059
100	5	5	0.048	0.051
100	10	5	0.056	0.057
100	20	10	0.051	0.051
b) Exponential error				
10	1	1	0.049	0.045
20	1	1	0.046	0.049
20	5	5	0.044	0.051
20	10	5	0.035	0.042
20	20	10	0.044	0.054
50	1	1	0.051	0.050
50	5	5	0.042	0.048
50	10	5	0.048	0.051
50	20	10	0.051	0.053
100	1	1	0.056	0.058
100	5	5	0.057	0.062
100	10	5	0.042	0.042
100	20	10	0.036	0.039

c) Cubed exponential error

10	1	1	0.046	0.055
20	1	1	0.061	0.064
20	5	5	0.043	0.048
20	10	5	0.050	0.065
20	20	10	0.029	0.036
50	1	1	0.046	0.044
50	5	5	0.047	0.046
50	10	5	0.054	0.055
50	20	10	0.042	0.045
100	1	1	0.048	0.047
100	5	5	0.050	0.048
100	10	5	0.052	0.058
100	20	10	0.063	0.056

Table A2. Simulation results for type I error in *F*-tests of significance of the global R^2 statistic for unstandardized **Y** variables generated with equal population variances. Headings as in Table A1.

n	*p*	*m*	Rejection rates	
			Parametric	Permutation
a) Normal error				
20	5	5	0.043	0.041
20	10	5	0.045	0.045
20	20	10	0.049	0.057
50	5	5	0.052	0.049
50	10	5	0.039	0.036
50	20	10	0.050	0.049
100	5	5	0.034	0.033
100	10	5	0.046	0.047
100	20	10	0.050	0.050
b) Exponential error				
20	5	5	0.069	0.054
20	10	5	0.066	0.045
20	20	10	0.073	0.052
50	5	5	0.051	0.043
50	10	5	0.056	0.047
50	20	10	0.065	0.063
100	5	5	0.049	0.045
100	10	5	0.047	0.045
100	20	10	0.036	0.035
c) Cubed exponential error				
20	5	5	0.141	0.054
20	10	5	0.161	0.045
20	20	10	0.228	0.051
50	5	5	0.103	0.041
50	10	5	0.139	0.058
50	20	10	0.182	0.049
100	5	5	0.084	0.046
100	10	5	0.142	0.051
100	20	10	0.154	0.045

Table A3. Simulation results for type I error in *F*-tests of significance of the global R^2 statistic for unstandardized **Y** variables with unequal population variances. Headings as in Table A1.

n	*p*	*m*	Rejection rates	
			Parametric	Permutation
a) Normal error				
20	5	5	0.071	0.038
20	10	5	0.094	0.046
20	20	10	0.100	0.054
50	5	5	0.094	0.050
50	10	5	0.084	0.052
50	20	10	0.093	0.044
100	5	5	0.101	0.068
100	10	5	0.100	0.057
100	20	10	0.112	0.056
b) Exponential error				
20	5	5	0.094	0.044
20	10	5	0.098	0.041
20	20	10	0.131	0.050
50	5	5	0.073	0.036
50	10	5	0.104	0.063
50	20	10	0.097	0.054
100	5	5	0.088	0.045
100	10	5	0.079	0.046
100	20	10	0.095	0.046
c) Cubed exponential error				
20	5	5	0.138	0.055
20	10	5	0.169	0.044
20	20	10	0.250	0.047
50	5	5	0.134	0.054
50	10	5	0.153	0.053
50	20	10	0.195	0.048
100	5	5	0.114	0.054
100	10	5	0.158	0.053
100	20	10	0.173	0.050

APPENDIX S2

SIMULATION RESULTS, TEST OF INDIVIDUAL CANONICAL AXES

The following tables of simulation results for the tests of individual canonical axes contain rejection rates after 1000 independent simulations. They should be read as follows.

• In the "Type I error" sections, all rejection rates are expected to be approximately equal or smaller than the significance level (0.05) used in the tests.

• In the simulations reported in the "Power and type I error" sections of Tables B1 and B2, as well as in Tables B3 to B7, 1 to 4 sets of linear relationships (called "blocks" in the column headings) have been generated between the response (**Y**) and explanatory (**X**) matrices. We want to know how much power the methods have to identify these relationships. Power is a value between 0 (no relationship detected, no power) and 1 (maximum power: the method has detected a relationships in all simulations). In Tables B4 ad B5, the "direct" and "reduced" permutation methods are compared for their power. In Tables B6 and B7, we will see the effect of non-normal error on the power of the tests.

• In the "Power and type I error" sections of Tables B1 and B2, as well as in Tables B3 to B7, there are more canonical axes than the number of built-in blocks; the simulation results (rejection rates) for these axes are in bold. Detecting one of these axes as significant is to make a type I error. A number of rejections of H_0 equal to or smaller than the significance level of 0.05, as found e.g. in Tables B2 and B3, is expected in valid tests at significance level 0.05. However, a value higher than 0.05 means that the method is invalid for testing these extra axes (e.g. in Table B1).

Table B1. Simulation results for the **simultaneous test** of all canonical axes. Shown in the table are the rejection rates of H_0 at the 5% significance level over 1000 simulations. Tests of significance involved 999 random permutations of the raw data using (a) statistic *F*1 or (b) statistic *F*2. Error level in the generation of the **y** variables (normal error): $c = 0.2$ in the *Power and type I error* sections of this table.

n	*p*	*m*	No.	No. **y**	No. can.	Rejection rates							
			blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
a) *F*1 statistic (equation 1)													
Type I error													
20	3	3	0	0	3	0.046	0.050	0.049					
100	3	3	0	0	3	0.050	0.035	0.043					
20	5	5	0	0	5	0.055	0.051	0.052	0.071	0.061			
100	5	5	0	0	5	0.048	0.058	0.055	0.043	0.052			
20	8	8	0	0	8	0.055	0.053	0.061	0.064	0.047	0.055	0.053	0.055
100	8	8	0	0	8	0.046	0.043	0.058	0.057	0.047	0.042	0.044	0.045
Power and type I error													
20	3	2	1	3	2	1.000	**0.799**						
100	3	2	1	3	2	1.000	**0.817**						
20	6	4	2	3	4	1.000	1.000	**0.997**	**0.910**				
100	6	4	2	3	4	1.000	1.000	**0.998**	**0.913**				
20	9	6	3	3	6	1.000	1.000	1.000	**1.000**	**0.998**	**0.946**		
100	9	6	3	3	6	1.000	1.000	1.000	**1.000**	**1.000**	**0.953**		
20	12	8	4	3	8	1.000	1.000	1.000	1.000	**1.000**	**1.000**	**1.000**	**0.988**
100	12	8	4	3	8	1.000	1.000	1.000	1.000	**1.000**	**1.000**	**1.000**	**0.977**

n	*p*	*m*	No.	No. **y**	No. can.	Rejection rates							
			blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
b) *F*2 statistic (equation 2)													
Type I error													
20	3	3	0	0	3	0.052	0.049	0.049					
100	3	3	0	0	3	0.051	0.036	0.043					
20	5	5	0	0	5	0.051	0.046	0.054	0.070	0.061			
100	5	5	0	0	5	0.048	0.058	0.054	0.043	0.052			
20	8	8	0	0	8	0.063	0.056	0.059	0.063	0.044	0.053	0.051	0.055
100	8	8	0	0	8	0.045	0.045	0.059	0.059	0.047	0.042	0.044	0.045
Power and type I error													
20	3	2	1	3	2	1.000	**0.799**						
100	3	2	1	3	2	1.000	**0.817**						
20	6	4	2	3	4	1.000	1.000	**0.997**	**0.910**				
100	6	4	2	3	4	1.000	1.000	**0.998**	**0.913**				
20	9	6	3	3	6	1.000	1.000	1.000	**1.000**	**0.998**	**0.946**		
100	9	6	3	3	6	1.000	1.000	1.000	**1.000**	**1.000**	**0.953**		
20	12	8	4	3	8	1.000	1.000	1.000	1.000	**1.000**	**1.000**	**1.000**	**0.988**
100	12	8	4	3	8	1.000	1.000	1.000	1.000	**1.000**	**1.000**	**1.000**	**0.977**

Notes: *n* = number of objects; *p* = number of response variables in **Y**; *m* = number of explanatory variables in **X**; No. blocks = number of blocks of **Y** variables linearly related to groups of 2 **X** variables; No. **y** per block = number of **Y** variables in each block; No. can. axes = number of resulting canonical axes.

Table B2. Simulation results for the **marginal test** of canonical eigenvalues: rejection rates of H_0 at the 5% significance level over 1000 simulations, normal error. Tests of significance involved 999 random permutations under the reduced model. Column headings: see notes, Table B1. The error level c = 0.2 in the "Power and type I error" sections.

n	*p*	*m*	No.	No. **y**	No. can.	Rejection rates							
			blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Type I error													
20	3	3	0	0	3	0.043	0.000	0.000					
100	3	3	0	0	3	0.050	0.000	0.000					
20	5	5	0	0	5	0.048	0.000	0.000	0.000	0.000			
100	5	5	0	0	5	0.051	0.000	0.000	0.000	0.000			
20	8	8	0	0	8	0.063	0.000	0.000	0.000	0.000	0.000	0.000	0.000
100	8	8	0	0	8	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Power and type I error													
20	1	2	1	1	1	1.000							
100	1	2	1	1	1	1.000							
20	2	2	1	2	2	1.000	**0.013**						
100	2	2	1	2	2	1.000	**0.022**						
20	3	2	1	3	2	1.000	**0.015**						
100	3	2	1	3	2	1.000	**0.014**						
20	2	4	2	1	2	1.000	0.999						
100	2	4	2	1	2	1.000	1.000						
20	4	4	2	2	4	1.000	1.000	**0.005**	**0.000**				
100	4	4	2	2	4	1.000	1.000	**0.003**	**0.000**				
20	6	4	2	3	4	1.000	1.000	**0.013**	**0.000**				
100	6	4	2	3	4	1.000	1.000	**0.011**	**0.000**				
20	3	6	3	1	3	1.000	1.000	0.981					
100	3	6	3	1	3	1.000	1.000	1.000					
20	6	6	3	2	6	1.000	1.000	0.998	**0.003**	**0.000**	**0.000**		
100	6	6	3	2	6	1.000	1.000	1.000	**0.005**	**0.000**	**0.000**		
20	9	6	3	3	6	1.000	1.000	1.000	**0.003**	**0.000**	**0.000**		
100	9	6	3	3	6	1.000	1.000	1.000	**0.007**	**0.000**	**0.000**		
20	4	8	4	1	4	1.000	1.000	1.000	0.687				
100	4	8	4	1	4	1.000	1.000	1.000	1.000				
20	8	8	4	2	8	1.000	1.000	1.000	0.930	**0.000**	**0.000**	**0.000**	**0.000**
100	8	8	4	2	8	1.000	1.000	1.000	1.000	**0.001**	**0.000**	**0.000**	**0.000**
20	12	8	4	3	8	1.000	1.000	1.000	0.985	**0.000**	**0.000**	**0.000**	**0.000**
100	12	8	4	3	8	1.000	1.000	1.000	1.000	**0.008**	**0.000**	**0.000**	**0.000**

Table B3. Simulation results for the **forward test** of canonical eigenvalues: rejection rates of H_0 at the 5% significance level over 1000 simulations, normal error. Tests of significance involved 999 random permutations under the reduced model. Column headings: see notes, Table B1. The error level c = 0.2 in the "Power and type I error" sections.

n	*p*	*m*	No.	No. **y**	No. can.	Rejection rates							
			blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Type I error													
20	3	3	0	0	3	0.042	0.000	0.000					
100	3	3	0	0	3	0.054	0.000	0.000					
20	5	5	0	0	5	0.047	0.000	0.000	0.000	0.000			
100	5	5	0	0	5	0.048	0.001	0.000	0.000	0.000			
20	8	8	0	0	8	0.057	0.001	0.000	0.000	0.000	0.000	0.000	0.000
100	8	8	0	0	8	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Power and type I error													
20	1	2	1	1	1	1.000							
100	1	2	1	1	1	1.000							
20	2	2	1	2	2	1.000	**0.013**						
100	2	2	1	2	2	1.000	**0.022**						
20	3	2	1	3	2	1.000	**0.016**						
100	3	2	1	3	2	1.000	**0.013**						
20	2	4	2	1	2	1.000	0.999						
100	2	4	2	1	2	1.000	1.000						
20	4	4	2	2	4	1.000	1.000	**0.007**	**0.000**				
100	4	4	2	2	4	1.000	1.000	**0.003**	**0.000**				
20	6	4	2	3	4	1.000	1.000	**0.015**	**0.000**				
100	6	4	2	3	4	1.000	1.000	**0.013**	**0.000**				
20	3	6	3	1	3	1.000	1.000	0.986					
100	3	6	3	1	3	1.000	1.000	1.000					
20	6	6	3	2	6	1.000	1.000	1.000	**0.004**	**0.000**	**0.000**		
100	6	6	3	2	6	1.000	1.000	1.000	**0.005**	**0.000**	**0.000**		
20	9	6	3	3	6	1.000	1.000	1.000	**0.006**	**0.000**	**0.000**		
100	9	6	3	3	6	1.000	1.000	1.000	**0.007**	**0.000**	**0.000**		
20	4	8	4	1	4	1.000	1.000	0.999	0.821				
100	4	8	4	1	4	1.000	1.000	1.000	1.000				
20	8	8	4	2	8	1.000	1.000	1.000	0.961	**0.000**	**0.000**	**0.000**	**0.000**
100	8	8	4	2	8	1.000	1.000	1.000	1.000	**0.001**	**0.000**	**0.000**	**0.000**
20	12	8	4	3	8	1.000	1.000	1.000	0.989	**0.003**	**0.000**	**0.000**	**0.000**
100	12	8	4	3	8	1.000	1.000	1.000	1.000	**0.011**	**0.000**	**0.000**	**0.000**

Table B4. Simulation results comparing permutation of the raw data (method = "direct") to permutation of the residuals of the reduced model (method = "reduced") for the **marginal test** of the canonical axes for different error levels *c* = {0.5, 0.8}, normal error. Rejection rates of H_0 at the 5% significance level over 1000 simulations. Tests of significance involved 999 random permutations. Column headings: see notes, Table B1.

n	*c*	No.	No. **y**	No. can.	Rejection rates							
		blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Method = "direct"												
20	0.5	1	3	2	1.000	**0.011**						
100	0.5	1	3	2	1.000	**0.005**						
20	0.5	2	3	4	1.000	0.860	**0.000**	**0.000**				
100	0.5	2	3	4	1.000	1.000	**0.001**	**0.000**				
20	0.5	3	3	6	1.000	0.954	0.268	**0.000**	**0.000**	**0.000**		
100	0.5	3	3	6	1.000	1.000	1.000	**0.000**	**0.000**	**0.000**		
20	0.5	4	3	8	1.000	0.972	0.583	0.011	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	1.000	1.000	1.000	0.999	**0.000**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.994	**0.015**						
100	0.8	1	3	2	1.000	**0.010**						
20	0.8	2	3	4	0.999	0.527	**0.002**	**0.000**				
100	0.8	2	3	4	1.000	1.000	**0.004**	**0.000**				
20	0.8	3	3	6	0.999	0.689	0.062	**0.000**	**0.000**	**0.000**		
100	0.8	3	3	6	1.000	1.000	0.988	**0.002**	**0.000**	**0.000**		
20	0.8	4	3	8	0.998	0.714	0.145	0.003	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	1.000	1.000	1.000	0.665	**0.000**	**0.000**	**0.000**	**0.000**
Method = "reduced"												
20	0.5	1	3	2	1.000	**0.022**						
100	0.5	1	3	2	1.000	**0.026**						
20	0.5	2	3	4	1.000	0.978	**0.006**	**0.000**				
100	0.5	2	3	4	1.000	1.000	**0.013**	**0.000**				
20	0.5	3	3	6	1.000	1.000	0.641	**0.003**	**0.000**	**0.000**		
100	0.5	3	3	6	1.000	1.000	1.000	**0.006**	**0.000**	**0.000**		
20	0.5	4	3	8	1.000	0.999	0.909	0.132	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	1.000	1.000	1.000	1.000	**0.002**	**0.000**	**0.000**	**0.000**

20	0.8	1	3	2	0.991	**0.017**						
100	0.8	1	3	2	1.000	**0.027**						
20	0.8	2	3	4	1.000	0.655	**0.002**	**0.000**				
100	0.8	2	3	4	1.000	1.000	**0.009**	**0.000**				
20	0.8	3	3	6	0.999	0.858	0.121	**0.000**	**0.000**	**0.000**		
100	0.8	3	3	6	1.000	1.000	1.000	**0.011**	**0.000**	**0.000**		
20	0.8	4	3	8	1.000	0.907	0.293	0.004	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	1.000	1.000	1.000	0.916	**0.004**	**0.000**	**0.000**	**0.000**

Table B5. Simulation results comparing the effects of different error levels $c = \{0.5, 0.8\}$, normal error, on the **forward test** of the canonical axes. Rejection rates of H_0 at the 5% significance level over 1000 simulations. Tests of significance involved 999 random permutations under the reduced model (method = “reduced”). Column headings: see notes, Table B1.

n	*c*	No.	No. **y**	No. can.	Rejection rates							
		blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Method = “reduced”												
20	0.5	1	3	2	1.000	**0.022**						
100	0.5	1	3	2	1.000	**0.025**						
20	0.5	2	3	4	1.000	0.976	**0.007**	**0.000**				
100	0.5	2	3	4	1.000	1.000	**0.011**	**0.000**				
20	0.5	3	3	6	1.000	1.000	0.673	**0.005**	**0.000**	**0.000**		
100	0.5	3	3	6	1.000	1.000	1.000	**0.005**	**0.000**	**0.000**		
20	0.5	4	3	8	1.000	0.996	0.905	0.189	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	1.000	1.000	1.000	1.000	**0.003**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.991	**0.016**						
100	0.8	1	3	2	1.000	**0.024**						
20	0.8	2	3	4	0.997	0.666	**0.004**	**0.000**				
100	0.8	2	3	4	1.000	1.000	**0.009**	**0.000**				
20	0.8	3	3	6	0.998	0.846	0.138	**0.002**	**0.000**	**0.000**		
100	0.8	3	3	6	1.000	1.000	1.000	**0.011**	**0.000**	**0.000**		
20	0.8	4	3	8	0.989	0.839	0.303	0.011	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	1.000	1.000	1.000	0.922	**0.005**	**0.000**	**0.000**	**0.000**

Table B6. Simulation results comparing the **marginal** and **forward** testing methods, under permutation of residuals (method = “reduced”), for the canonical axes using **exponential error**, for different error levels *c* = {0.2, 0.5, 0.8}. Rejection rates of H_0 at the 5% significance level over 1000 simulations. Tests of significance involved 999 random permutations. Column headings: see notes, Table B1.

n	*c*	No.	No. **y**	No. can.	Rejection rates							
		blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Marginal test												
20	0.2	1	3	2	1.000	**0.020**						
100	0.2	1	3	2	1.000	**0.018**						
20	0.2	2	3	4	1.000	1.000	**0.015**	**0.000**				
100	0.2	2	3	4	1.000	1.000	**0.010**	**0.000**				
20	0.2	3	3	6	1.000	1.000	1.000	**0.006**	**0.000**	**0.000**		
100	0.2	3	3	6	1.000	1.000	1.000	**0.004**	**0.000**	**0.000**		
20	0.2	4	3	8	1.000	1.000	1.000	0.975	**0.002**	**0.000**	**0.000**	**0.000**
100	0.2	4	3	8	1.000	1.000	1.000	1.000	**0.010**	**0.000**	**0.000**	**0.000**
20	0.5	1	3	2	1.000	**0.035**						
100	0.5	1	3	2	1.000	**0.026**						
20	0.5	2	3	4	1.000	0.951	**0.016**	**0.000**				
100	0.5	2	3	4	1.000	1.000	**0.008**	**0.000**				
20	0.5	3	3	6	1.000	1.000	0.577	**0.006**	**0.000**	**0.000**		
100	0.5	3	3	6	1.000	1.000	1.000	**0.006**	**0.000**	**0.000**		
20	0.5	4	3	8	1.000	1.000	0.880	0.115	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	1.000	1.000	1.000	1.000	**0.004**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.976	**0.018**						
100	0.8	1	3	2	1.000	**0.020**						
20	0.8	2	3	4	0.995	0.592	**0.008**	**0.000**				
100	0.8	2	3	4	1.000	1.000	**0.003**	**0.000**				
20	0.8	3	3	6	0.996	0.809	0.110	**0.001**	**0.000**	**0.000**		
100	0.8	3	3	6	1.000	1.000	0.997	**0.004**	**0.000**	**0.000**		
20	0.8	4	3	8	0.996	0.841	0.253	0.004	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	1.000	1.000	1.000	0.883	**0.004**	**0.000**	**0.000**	**0.000**

Forward test

20	0.2	1	3	2	1.000	**0.022**						
100	0.2	1	3	2	1.000	**0.018**						
20	0.2	2	3	4	1.000	1.000	**0.020**	**0.000**				
100	0.2	2	3	4	1.000	1.000	**0.010**	**0.000**				
20	0.2	3	3	6	1.000	1.000	1.000	**0.008**	**0.000**	**0.000**		
100	0.2	3	3	6	1.000	1.000	1.000	**0.003**	**0.000**	**0.000**		
20	0.2	4	3	8	1.000	1.000	1.000	0.987	**0.004**	**0.000**	**0.000**	**0.000**
100	0.2	4	3	8	1.000	1.000	1.000	1.000	**0.008**	**0.000**	**0.000**	**0.000**
20	0.5	1	3	2	1.000	**0.036**						
100	0.5	1	3	2	1.000	**0.026**						
20	0.5	2	3	4	1.000	0.954	**0.017**	**0.000**				
100	0.5	2	3	4	1.000	1.000	**0.008**	**0.000**				
20	0.5	3	3	6	1.000	0.997	0.598	**0.007**	**0.000**	**0.000**		
100	0.5	3	3	6	1.000	1.000	1.000	**0.006**	**0.000**	**0.000**		
20	0.5	4	3	8	1.000	0.995	0.873	0.146	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	1.000	1.000	1.000	1.000	**0.005**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.975	**0.016**						
100	0.8	1	3	2	1.000	**0.017**						
20	0.8	2	3	4	0.991	0.607	**0.009**	**0.000**				
100	0.8	2	3	4	1.000	1.000	**0.003**	**0.000**				
20	0.8	3	3	6	0.987	0.789	0.132	**0.000**	**0.000**	**0.000**		
100	0.8	3	3	6	1.000	1.000	0.997	**0.003**	**0.000**	**0.000**		
20	0.8	4	3	8	0.972	0.751	0.258	0.011	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	1.000	1.000	1.000	0.891	**0.002**	**0.000**	**0.000**	**0.000**

Table B7. Simulation results comparing the **marginal** and **forward** testing methods, under permutation of residuals (method = “reduced”), for the canonical axes using **cubed exponential error**, for different error levels c = {0.2, 0.5, 0.8}. Rejection rates of H_0 at the 5% significance level over 1000 simulations. Tests of significance involved 999 random permutations. Column headings: see notes, Table B1.

n	*c*	No.	No. **y**	No. can.	Rejection rates							
		blocks	per block	axes	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
Marginal test												
20	0.2	1	3	2	0.251	**0.003**						
100	0.2	1	3	2	0.472	**0.006**						
20	0.2	2	3	4	0.135	0.009	**0.000**	**0.000**				
100	0.2	2	3	4	0.343	0.030	**0.000**	**0.000**				
20	0.2	3	3	6	0.093	0.004	0.001	**0.000**	**0.000**	**0.000**		
100	0.2	3	3	6	0.236	0.018	0.001	**0.000**	**0.000**	**0.000**		
20	0.2	4	3	8	0.080	0.001	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.2	4	3	8	0.155	0.009	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
20	0.5	1	3	2	0.091	**0.001**						
100	0.5	1	3	2	0.112	**0.000**						
20	0.5	2	3	4	0.064	0.001	**0.000**	**0.000**				
100	0.5	2	3	4	0.079	0.000	**0.000**	**0.000**				
20	0.5	3	3	6	0.055	0.000	0.000	**0.000**	**0.000**	**0.000**		
100	0.5	3	3	6	0.057	0.002	0.000	**0.000**	**0.000**	**0.000**		
20	0.5	4	3	8	0.043	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	0.057	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.058	**0.000**						
100	0.8	1	3	2	0.074	**0.000**						
20	0.8	2	3	4	0.063	0.001	**0.000**	**0.000**				
100	0.8	2	3	4	0.058	0.000	**0.000**	**0.000**				
20	0.8	3	3	6	0.053	0.001	0.000	**0.000**	**0.000**	**0.000**		
100	0.8	3	3	6	0.067	0.000	0.000	**0.000**	**0.000**	**0.000**		
20	0.8	4	3	8	0.053	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	0.069	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**

Forward test

20	0.2	1	3	2	0.250	**0.003**						
100	0.2	1	3	2	0.469	**0.007**						
20	0.2	2	3	4	0.121	0.009	**0.000**	**0.000**				
100	0.2	2	3	4	0.343	0.030	**0.000**	**0.000**				
20	0.2	3	3	6	0.081	0.004	0.001	**0.000**	**0.000**	**0.000**		
100	0.2	3	3	6	0.226	0.017	0.001	**0.000**	**0.000**	**0.000**		
20	0.2	4	3	8	0.052	0.003	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.2	4	3	8	0.140	0.009	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
20	0.5	1	3	2	0.083	**0.001**						
100	0.5	1	3	2	0.113	**0.000**						
20	0.5	2	3	4	0.054	0.001	**0.000**	**0.000**				
100	0.5	2	3	4	0.081	0.000	**0.000**	**0.000**				
20	0.5	3	3	6	0.057	0.000	0.000	**0.000**	**0.000**	**0.000**		
100	0.5	3	3	6	0.058	0.002	0.000	**0.000**	**0.000**	**0.000**		
20	0.5	4	3	8	0.045	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.5	4	3	8	0.055	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
20	0.8	1	3	2	0.058	**0.000**						
100	0.8	1	3	2	0.075	**0.000**						
20	0.8	2	3	4	0.058	0.001	**0.000**	**0.000**				
100	0.8	2	3	4	0.060	0.000	**0.000**	**0.000**				
20	0.8	3	3	6	0.048	0.005	0.000	**0.000**	**0.000**	**0.000**		
100	0.8	3	3	6	0.065	0.000	0.000	**0.000**	**0.000**	**0.000**		
20	0.8	4	3	8	0.057	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**
100	0.8	4	3	8	0.066	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**

Supplementary information to:

Legendre, P., Oksanen, J. & ter Braak, C. J. F. (2011) Testing the significance of canonical axes in redundancy analysis. *Methods in Ecology & Evolution*, **2**, 269–277.

DATA S1

DESCRIPTION OF THE MARGINAL AND FORWARD METHODS

This section contains R-language functions describing the marginal and forward methods for testing the canonical axes in simple redundancy analysis (RDA) without covariables. These functions are also found in the folder of R functions provided in the Supporting Information.

Marginal test

```
'marginal.test' <-
   function(Y,X,nperm=199,seed=NULL,scaleY=FALSE,verbose=FALSE)
#
# Test of canonical eigenvalues in simple RDA using the marginal method.
# In this method, the residual SS in the denominator of F is the same for the
# test of all canonical axes: the residual SS of the regression of Y on X.
#
# This is a simple explicit program, without shortcuts nor compiled
# permutation function. Its aim is to unambiguously describe the marginal
# testing method. The code is interspersed with comments to explain the
# computation steps. This function is not intended for routine testing of
# canonical eigenvalues, although it produces correct, publishable results.
#
# Parameters:
#    Y: response data matrix
#    X: explanatory data matrix; no factors are allowed
#    nperm: number of permutations
#    seed: seed for random number generator, used by the permutation function
#          sample(). If seed=NULL (default), a random integer is drawn as the
#          seed for the run. It will be reset to that value before the test of
#          each canonical axis. All axes are thus tested using the same set of
#          permutations. A fixed value, e.g. seed=12345, can be givenn by the
#          user to compare the results of this function with that of other
#          functions where the seed can also be set at run time.
#    scaleY = TRUE : standardize the Y variables
#           = FALSE: the Y variables are centered on their means
#    verbose = TRUE: print intermediate F and F.perm results
#
# License: GPL-2
# Authors: Pierre Legendre and Jari Oksanen, 2010
{
   if(nperm > 5) verbose <- FALSE   # Modify this limit as needed #
   if(length(seed)==0) seed <- ceiling(runif(1,max=10000))
   cat("seed =",seed,'\n')
   a <- system.time({             # How much time needed for the test?
   epsilon <- sqrt(.Machine$double.eps)
#
# Center or standardize the Y variables, standardize the X variables
   Y <- scale(as.matrix(Y), center=TRUE, scale=scaleY)
   X <- scale(as.matrix(X), center=TRUE, scale=TRUE)
   n <- nrow(Y)
   SS.Y <- sum(Y^2)
#
# Compute the rank of X
   m <- length(which(svd(cov(X))$d > epsilon))
#
# RDA consists of 2 steps: regression, then PCA of the table of fitted
# values by eigen decomposition of the matrix of Sums of Squares and Cross
# Products (SSCP), which is cov*(n-1)
   qr.X <- qr(X)                   # We use QR decomposition for efficiency
   Yhat <- qr.fitted(qr.X, Y)      # Faster than: Yhat<-fitted.values(lm(Y~X))
   SS.Yhat <- sum(Yhat^2)
   Yhat.eig <- svd(Yhat, nv=0)     # svd decomposition of Yhat
   eig.values <- Yhat.eig$d^2      # = eigen(cov( t(Yhat)%*%Yhat ))$values
   k <- sum(eig.values > epsilon)  # Number of canonical eigenvalues
   axes <- Yhat.eig$u[,1:k]        # Canonical axes = f(X)
#
# A section could be added here to compute the RDA R-square (SS.Yhat/SS.Y)
# and test its significance.
# A section could be added here to produce a biplot with scaling type 1 or 2.
#
# Test the significance of individual canonical axes
   #
   # No covariables: test canonical axis 1 by permutation of the raw data.
   # Test of axis 1: no residuals to be computed on previously tested axes
   F <- eig.values[1] / (SS.Y - SS.Yhat)
   Fstat <- F*(n-1-m)        # Start assembling vector 'Fstat'
   #
   set.seed(seed)
   nGE <- 1   # Hope correction: count ‘Fstat’ in the reference distribution
   for(iperm in 1:nperm) {
      Y.perm <- Y[sample(n),]
      SS.Y.perm <- SS.Y      # Same SS for Y and Y.perm after permuting Y rows
      #
      # Re-use the qr.X: does not change during permutations
      Yhat.perm <- qr.fitted(qr.X, Y.perm)
      SS.Yhat.perm <- sum(Yhat.perm^2)
      Yhat.perm.eig <- svd(Yhat.perm, nv=0, nu=0)$d^2
      #
      F.perm <- Yhat.perm.eig[1] / (SS.Y.perm - SS.Yhat.perm)
      if(F.perm >= F) nGE <- nGE+1
      if(verbose) cat("Axis 1   :   F =",F,"  F.perm =",F.perm,'\n')
      }
   prob <- nGE/(nperm+1)     # Start assembling vector 'prob'

   # Test the following canonical axes by permuting residuals of reduced model
   if(k > 1) {
   for(j in 2:k) {
      # Compute the F statistic
      F <- eig.values[j] / (SS.Y - SS.Yhat)
      Fstat <- c(Fstat, F*(n-1-m))     # Attach the new F value to 'Fstat'
      #
      # Compute reduced model residuals of Y on the previously tested axes
      qr.prev  <- qr(axes[,1:(j-1)])
      Y.fit <- qr.fitted(qr.prev, Y)
      Y.res <- qr.resid(qr.prev, Y)
      #
      # Compute residuals of X on previously tested axes for partial RDA below
      qr.X.res <- qr(qr.resid(qr.prev, X))
      #
      set.seed(seed)
      nGE <- 1   # Hope correction
      if(verbose) cat('\n')
      for(iperm in 1:nperm) {
         # Create permuted Y and compute its sum of squares (SS)
         Y.perm <- Y.fit + Y.res[sample(n),]
         SS.Y.perm <- sum(Y.perm^2)    # Not same SS as in the unpermuted data
         #
         # The j-th eigenvalue is the first eigenvalue of the partial RDA
         # of Y.perm by X residualized onthe previous axes[,1:(j-1)].
         # qr.X.res was computed before the permutation loop.
         Yhat.perm <- qr.fitted(qr.X.res, Y.perm)
         Yhat.perm.eig1 <- svd(Yhat.perm, nv=0, nu=0)$d[1]^2
         #
         # For denominator of the F-statistic: RDA of Y.perm on X.
         # qr.X was computed near the beginning of the function.
         YhatTot.perm <- qr.fitted(qr.X, Y.perm)
         SS.YhatTot.perm <- sum(YhatTot.perm^2)
         #
         # Compute the F-statistic under permutation for the tested eigenvalue
         F.perm <- Yhat.perm.eig1 / (SS.Y.perm - SS.YhatTot.perm)
         #
         if(F.perm >= F) nGE <- nGE+1
         if(verbose) cat("Axis [",j,"]:  F =",F,"  F.perm =",F.perm,'\n')
         }
      prob <- c(prob, nGE/(nperm+1))   # Attach the new probability to 'prob'
      }
      }
#
   })
   a[3] <- sprintf("%2f",a[3])
   cat("Computation time =",a[3]," sec",'\n')
#
# Output the  results
# The eigenvalues of the covariance matrix = (eigenvalues computed here)/(n-1)
out <- cbind(eig.values[1:k]/(n-1), Fstat, prob)
#
rownames(out) <- rownames(out, do.NULL=FALSE, prefix="Axis.")
colnames(out) <- c("Eigenvalue", "F.marginal", "P.marginal")
out
}
```

Forward test

```
'forward.test' <-
   function(Y, X, nperm=199, seed=NULL, scaleY=FALSE, verbose=FALSE)
#
# Test of canonical eigenvalues in simple RDA using the forward method.
# In this method, the residual SS in the denominator of F differs from axis
# to axis: it is the residual SS of the regression of Y on canonical axes
# from 1 to the one being tested.
#
# This is a simple explicit program, without shortcuts nor compiled
# permutation function. Its aim is to unambiguously describe the stepwise
# testing method. The code is interspersed with comments to explain the
# computation steps. This function is not intended for routine testing of
# canonical eigenvalues, although it produces correct, publishable results.
#
# Parameters:
#    Y: response data matrix
#    X: explanatory data matrix
#    nperm: number of permutations
#    seed: seed for random number generator, used by the permutation function
#          sample(). If seed=NULL (default), a random integer is drawn as the
#          seed for the run. It will be reset to that value before the test of
#          each canonical axis. All axes are thus tested using the same set of
#          permutations. A fixed value, e.g. seed=12345, can be givenn by the
#          user to compare the results of this function with that of other
#          functions where the seed can also be set at run time.
#    scaleY = TRUE : standardize the Y variables
#           = FALSE: the Y variables are centered on their means
#    verbose = TRUE: print intermediate F and F.perm results
#
# License: GPL-2
# Authors: Pierre Legendre, Cajo J. F. ter Braak and Jari Oksanen, 2010
{
   if(nperm > 5) verbose <- FALSE   # Modify this limit as needed #
   if(length(seed)==0) seed <- ceiling(runif(1,max=10000))
   cat("seed =",seed,'\n')
   a <- system.time({               # How much time needed for the test?
   epsilon <- sqrt(.Machine$double.eps)
#
# Center or standardize the Y variables, standardize the X variables
   Y <- scale(as.matrix(Y), center=TRUE, scale=scaleY)
   X <- scale(as.matrix(X), center=TRUE, scale=TRUE)
   n <- nrow(Y)
   SS.Y <- sum(Y^2)
#
# Compute the rank of X
   m <- sum(svd(X, nv = 0, nu = 0)$d^2 > epsilon)
#
# RDA consists of 2 steps: regression, done with QR to avoid overhead of lm(),
# then PCA of the table of fitted values by decomposition of the matrix of
# Sums of Squares and Cross Products (SSCP), which is the same as cov*(n-1).
   Q <- qr(X)                      # We use QR decomposition for efficiency
   Yhat <- qr.fitted(Q, Y)         # Faster than: Yhat<-fitted.values(lm(Y~X))
   SS.Yhat <- sum(Yhat^2)
   Yhat.eig <- svd(Yhat, nv=0)     # svd decomposition of Yhat
   eig.values <- Yhat.eig$d^2      # = eigen(cov( t(Yhat)%*%Yhat ))$values
   k <- sum(eig.values > epsilon)  # Number of canonical eigenvalues
   axes <- Yhat.eig$u[,1:k]        # Canonical axes = f(X)
#
# A section could be added here to compute the RDA R-square (SS.Yhat/SS.Y)
# and test its significance (F-test).
# A section could be added here to produce a biplot with scaling type 1 or 2.
#
# Test the significance of individual canonical axes
   #
   # Test of axis 1: no residuals to be computed on previously tested axes
   # No covariables: test canonical axis 1 by permutation of the raw data.
   F <- eig.values[1] / (SS.Y - eig.values[1])
   Fstat <- F*(n-1-m)        # Start assembling vector 'Fstat'
   sum.eigval <- eig.values[1]
   #
   set.seed(seed)
   nGE <- 1   # Hope correction: count 'Fstat' in the reference distribution
   for(iperm in 1:nperm) {
      Y.perm <- Y[sample(n),]
      SS.Y.perm <- SS.Y      # Same SS for Y and Y.perm after permuting Y rows
      # Re-use the QR decomposition of X: does not change in permutations
      Yhat.perm <- qr.fitted(Q, Y.perm)
      Yhat.perm.eig <- svd(Yhat.perm, nv=0, nu=0)$d^2
      eig.value.perm <- Yhat.perm.eig[1]
      #
      F.perm <- eig.value.perm / (SS.Y.perm - eig.value.perm)
      if(F.perm >= F) nGE <- nGE+1
      if(verbose) cat("Axis 1   :   F =",F,"  F.perm =",F.perm,'\n')
      }
   prob <- nGE/(nperm+1)     # Start assembling vector 'prob'

   # Test the following canonical axes by permuting residuals of reduced model
   if(k > 1) {
   for(j in 2:k) {
      sum.eigval <- sum.eigval + eig.values[j]
      F <- eig.values[j] / (SS.Y - sum.eigval)
      Fstat <- c(Fstat, F*(n-1-m))   # Attach the new F value to 'Fstat'
      #
      # Compute reduced model residuals of Y on the previously tested axes
      qr.prev  <- qr(axes[,1:(j-1)])
      Y.fit <- qr.fitted(qr.prev, Y)
      Y.res <- qr.resid(qr.prev, Y)
      qr.X.res <- qr(qr.resid(qr.prev, X))

      set.seed(seed)
      nGE <- 1   # Hope correction
      if(verbose) cat('\n')
      for(iperm in 1:nperm) {
         # Create permuted Y and compute its sum of squares (SS)
         Y.perm <- Y.fit + Y.res[sample(n),]
         SS.Y.perm <- sum(Y.perm^2)   # Not same SS as in the unpermuted data
         #
         # The j-th eigenvalue is the first eigenvalue of the partial RDA
         # of Y.perm by X in the presence of the previous axes[,1:(j-1)].
         # Prior to the permutation loop, X has been residualized on the
         # previous axes.
         Yhat.perm <- qr.fitted(qr.X.res, Y.perm)
         Yhat.perm.eig1 <- svd(Yhat.perm, nv=0, nu=0)$d[1]^2
         #
         # The fitted values of Y.perm on the previous axes [1:(j-1)]
         # provide the first [1:(j-1)] constrained eigenvalues for the
         # denominator of F. We sum them, then we add the j-th eigenvalue
         # 'Yhat.perm.eig1' computed above.
         sum.eig.1toj.perm <- sum(qr.fitted(qr.prev, Y.perm)^2)+Yhat.perm.eig1
         #
         # Compute the F-statistic under permutation for the tested eigenvalue
         F.perm <- Yhat.perm.eig1 / (SS.Y.perm - sum.eig.1toj.perm)
         if(F.perm >= F) nGE <- nGE+1
         if(verbose) cat("Axis [",j,"]:  F =",F,"  F.perm =",F.perm,'\n')
         }
      prob <- c(prob, nGE/(nperm+1))   # Attach the new probability to 'prob'
      }
      }
#
  })
  a[3] <- sprintf("%2f",a[3])
  cat("Computation time =",a[3]," sec",'\n')
#
# Output the results.
# The eigenvalues of the covariance matrix = (eigenvalues computed here)/(n-1)
out <- cbind(eig.values[1:k]/(n-1), Fstat, prob)
#
rownames(out) <- rownames(out, do.NULL=FALSE, prefix="Axis.")
colnames(out) <- c("Eigenvalue", "F.forward", "P.forward")
out
}
```

* * * * * * * * * *

DATA S2

R FUNCTIONS FOR TESTING THE CANONICAL AXES IN PARTIAL RDA

Two functions are available in the R function folder to test the canonical axes in partial RDA, which is RDA in the presence of a matrix of covariables **W**.

• The function test.axes.canoco() follows the programming approach of the program Canoco (ter Braak and Šmilauer 2002); it uses the first approach described in the section "Partial RDA" of the paper. A first partial RDA is computed to obtain the canonical eigenvalues and the reference values of the *F*-statistics for the marginal and forward methods. Calculation of the partial RDA uses the residuals of **Y** on **W**, $\mathbf{Y}_{\mathrm{res}|\mathbf{W}}$, and the residuals of **X** on **W**, $\mathbf{X}_{\mathrm{res}|\mathbf{W}}$. Then, for each axis, a new partial RDA is computed involving matrix W incremented with addition of the previously tested axes (no axis is added for the test of the first axis), and the first axis of that partial RDA is tested for significance (that axis being the axis currently under test). The partial RDA function is called again after each permutation of the residuals of the reduced model.

• The function test.axes.cov() uses the second approach described in the section "Partial RDA" of the paper, where the RDA of **Y** by $\mathbf{X}_{res|\mathbf{W}}$ is computed to produce the canonical eigenvalues and test their significance. The canonical eigenvectors, as well as the ordination axes for the partial object scores and the fitted object scores, could also be computed in this approach; they are not in this demonstration function.

DATA S3

SIMULATION METHOD, TEST OF THE GLOBAL RDA R-SQUARE

This section contains the scripts and the parameter files used to carry out the simulations for the global tests of significance of the canonical relationship in RDA reported in the paper. The R simulation functions are in the function folder.

Simulations for results presented in Appendix S1, Tables A1 to A3

# Script to launch the simulations
# Load the following functions to the R window: run.simul.miller(), simul.miller() and generate.YX(). These three functions are found in the file simul.miller.R.

```
param <- read.table(file.choose())   # Read the file of simulation parameters
simul.out.11 = run.simul.miller(param, nruns=9, ierr=1, sd.max=1, scaleY=TRUE,
perm=TRUE)
simul.out.12 = run.simul.miller(param, nruns=9, ierr=2, sd.max=1, scaleY=TRUE,
perm=TRUE)
simul.out.13 = run.simul.miller(param, nruns=9, ierr=3, sd.max=1, scaleY=TRUE,
perm=TRUE)
simul.out.21 = run.simul.miller(param, nruns=9, ierr=1, sd.max=1,
scaleY=FALSE, perm=TRUE)
simul.out.22 = run.simul.miller(param, nruns=9, ierr=2, sd.max=1,
scaleY=FALSE, perm=TRUE)
simul.out.23 = run.simul.miller(param, nruns=9, ierr=3, sd.max=1,
scaleY=FALSE, perm=TRUE)
simul.out.31 = run.simul.miller(param, nruns=9, ierr=1, sd.max=10,
scaleY=FALSE, perm=TRUE)
simul.out.32 = run.simul.miller(param, nruns=9, ierr=2, sd.max=10,
scaleY=FALSE, perm=TRUE)
simul.out.33 = run.simul.miller(param, nruns=9, ierr=3, sd.max=10,
scaleY=FALSE, perm=TRUE)
```

# File of simulation parameters

```
      nsim  nperm n     p     m
Run1  1000  999   10    1     1
Run2  1000  999   20    1     1
Run3  1000  999   20    5     5
Run4  1000  999   20    10    5
Run5  1000  999   20    20    10
Run6  1000  999   50    1     1
Run7  1000  999   50    5     5
Run8  1000  999   50    10    5
Run9  1000  999   50    20    10
Run10 1000  999   100   1     1
Run11 1000  999   100   5     5
Run12 1000  999   100   10    5
Run13 1000  999   100   20    10
```

DATA S4

SIMULATION METHODS, TEST OF INDIVIDUAL CANONICAL AXES

This section contains the scripts and the parameter files used to carry out the simulations for the tests of significance of individual eigenvalues in RDA reported in the paper. The R simulation functions are in the function folder.

Simultaneous test of the canonical axes

```
# Simulations for the results presented in Appendix S2, Table B1
```

```
# Script to launch the simulations
# Load the following functions: run.simul.simultaneous(), generate.data() and
simul.test.eigval.simultaneous()

param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul.simultaneous(param, nruns=14)
out$F1                              # Results for statistic F1
out$F2                              # Results for statistic F2
```

```
# File of simulation parameters
```

```
      nsim  nperm n     p     m     block pp
Run1  1000  999   20    3     3     0     0
Run2  1000  999   100   3     3     0     0
Run3  1000  999   20    5     5     0     0
Run4  1000  999   100   5     5     0     0
Run5  1000  999   20    8     8     0     0
Run6  1000  999   100   8     8     0     0
Run7  1000  999   20    3     2     1     3
Run8  1000  999   100   3     2     1     3
Run9  1000  999   20    6     4     2     3
Run10 1000  999   100   6     4     2     3
Run11 1000  999   20    9     6     3     3
Run12 1000  999   100   9     6     3     3
Run13 1000  999   20    12    8     4     3
Run14 1000  999   100   12    8     4     3
```

Marginal and forward tests of the canonical axes

```
# Simulations for results presented in Appendix S2, Tables B2 and B3

# Load the following functions to the R window: run.simul(), generate.data(), simul.test.eigval(),
test.eigval() and forward.test()

param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul(param, nruns=30, ierr=1, direct=FALSE)
out$outJ                    # Results for the marginal test (Jari Oksanen method)
out$outC                    # Results for forward test (Cajo ter Braak method)
```

```
# File of simulation parameters
```

```
      nsim  nperm n     p     m     block pp    beta.err
Run1  1000  999   20    3     3     0     0     0.2
Run2  1000  999   100   3     3     0     0     0.2
Run3  1000  999   20    5     5     0     0     0.2
Run4  1000  999   100   5     5     0     0     0.2
Run5  1000  999   20    8     8     0     0     0.2
Run6  1000  999   100   8     8     0     0     0.2
Run7  1000  999   20    1     2     1     1     0.2
Run8  1000  999   100   1     2     1     1     0.2
Run9  1000  999   20    2     2     1     2     0.2
Run10 1000  999   100   2     2     1     2     0.2
Run11 1000  999   20    3     2     1     3     0.2
Run12 1000  999   100   3     2     1     3     0.2
Run13 1000  999   20    2     4     2     1     0.2
Run14 1000  999   100   2     4     2     1     0.2
Run15 1000  999   20    4     4     2     2     0.2
Run16 1000  999   100   4     4     2     2     0.2
Run17 1000  999   20    6     4     2     3     0.2
Run18 1000  999   100   6     4     2     3     0.2
Run19 1000  999   20    3     6     3     1     0.2
Run20 1000  999   100   3     6     3     1     0.2
Run21 1000  999   20    6     6     3     2     0.2
Run22 1000  999   100   6     6     3     2     0.2
Run23 1000  999   20    9     6     3     3     0.2
Run24 1000  999   100   9     6     3     3     0.2
Run25 1000  999   20    4     8     4     1     0.2
Run26 1000  999   100   4     8     4     1     0.2
Run27 1000  999   20    8     8     4     2     0.2
Run28 1000  999   100   8     8     4     2     0.2
Run29 1000  999   20    12    8     4     3     0.2
Run30 1000  999   100   12    8     4     3     0.2
```

```
# Simulations for results presented in Appendix S2, Table B4, method="direct"

# Script to launch the simulations
# Load the following functions to the R window: run.simul(), generate.data(), simul.test.eigval(),
test.eigval() and forward.test()

param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul(param, nruns=16, ierr=1, direct=TRUE)
out$outJ                           # Results for the marginal test (Jari Oksanen method)
# Discard out$outC: results for forward test, permutation of residuals of reduced model
```

```
# Simulations for results presented in Appendix S2, Tables B4 and B5, method="reduced"

# Script to launch the simulations
# Load the following functions to the R window: run.simul(), generate.data(), simul.test.eigval(),
test.eigval() and forward.test()

param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul(param, nruns=16, ierr=1, direct=FALSE)
out$outJ                           # Results for the marginal test (Jari Oksanen method)
out$outC                           # Results for forward test (Cajo ter Braak method)
```

```
# File of simulation parameters

      nsim  nperm n     p     m     block pp    beta.err
Run1  1000  999   20    3     2     1     3     0.5
Run2  1000  999   100   3     2     1     3     0.5
Run3  1000  999   20    6     4     2     3     0.5
Run4  1000  999   100   6     4     2     3     0.5
Run5  1000  999   20    9     6     3     3     0.5
Run6  1000  999   100   9     6     3     3     0.5
Run7  1000  999   20    12    8     4     3     0.5
Run8  1000  999   100   12    8     4     3     0.5
Run9  1000  999   20    3     2     1     3     0.8
Run10 1000  999   100   3     2     1     3     0.8
Run11 1000  999   20    6     4     2     3     0.8
Run12 1000  999   100   6     4     2     3     0.8
Run13 1000  999   20    9     6     3     3     0.8
Run14 1000  999   100   9     6     3     3     0.8
Run15 1000  999   20    12    8     4     3     0.8
Run16 1000  999   100   12    8     4     3     0.8
```

```
# Simulations for results presented in Appendix S2, Tables B6
# Script to launch the simulations
# Load the following functions to the R window: run.simul(), generate.data(), simul.test.eigval(),
test.eigval() and forward.test()
param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul(param, nruns=24, ierr=2, direct= FALSE)
out$outJ                        # Results for the marginal test (Jari Oksanen method)
out$outC                        # Results for forward test (Cajo ter Braak method)
```

```
# Simulations for results presented in Appendix S2, Tables B7
# Script to launch the simulations
# Load the following functions to the R window: run.simul(), generate.data(), simul.test.eigval(),
test.eigval() and forward.test()
param <- read.table(file.choose())   # Read the file of simulation parameters
out <- run.simul(param, nruns=24, ierr=3, direct=FALSE)
out$outJ                        # Results for the marginal test (Jari Oksanen method)
out$outC                        # Results for forward test (Cajo ter Braak method)
```

```
# File of simulation parameters
```

```
      nsim  nperm n     p     m     block pp    beta.err
Run1  1000  999   20    3     2     1     3     0.2
Run2  1000  999   100   3     2     1     3     0.2
Run3  1000  999   20    6     4     2     3     0.2
Run4  1000  999   100   6     4     2     3     0.2
Run5  1000  999   20    9     6     3     3     0.2
Run6  1000  999   100   9     6     3     3     0.2
Run7  1000  999   20    12    8     4     3     0.2
Run8  1000  999   100   12    8     4     3     0.2
Run9  1000  999   20    3     2     1     3     0.5
Run10 1000  999   100   3     2     1     3     0.5
Run11 1000  999   20    6     4     2     3     0.5
Run12 1000  999   100   6     4     2     3     0.5
Run13 1000  999   20    9     6     3     3     0.5
Run14 1000  999   100   9     6     3     3     0.5
Run15 1000  999   20    12    8     4     3     0.5
Run16 1000  999   100   12    8     4     3     0.5
Run17 1000  999   20    3     2     1     3     0.8
Run18 1000  999   100   3     2     1     3     0.8
Run19 1000  999   20    6     4     2     3     0.8
Run20 1000  999   100   6     4     2     3     0.8
Run21 1000  999   20    9     6     3     3     0.8
Run22 1000  999   100   9     6     3     3     0.8
Run23 1000  999   20    12    8     4     3     0.8
Run24 1000  999   100   12    8     4     3     0.8
```