Journal of Biogeography (*J. Biogeogr.*) (2006) **33**, 924–935

ORIGINAL ARTICLE

A new algorithm to calculate the nestedness temperature of presence–absence matrices

Miguel A. Rodríguez-Gironés[1]* and Luis Santamaría[2]

[1]*Estación Experimental de Zonas Áridas – CSIC, c/General Segura 1, Almeria, Spain,* [2]*Mediterranean Institute for Advanced Studies (IMEDEA, CSIC-UIB), c/Miquel Marquès 21, Esporles, Mallorca, Balearic Islands, Spain*

*Correspondence: Miguel A. Rodríguez-Gironés, Estación Experimental de Zonas Áridas – CSIC, c/General Segura 1, E-04001 Almeria, Spain.
E-mail: rgirones@eeza.csic.es

ABSTRACT

Aim The nestedness temperature of presence–absence matrices is currently calculated with the nestedness temperature calculator (NTC). In the algorithm implemented by the NTC: (1) the line of perfect order is not uniquely defined, (2) rows and columns are reordered in such a way that the packed matrix is not the one with the lowest temperature, and (3) the null model used to determine the probabilities of finding random matrices with the same or lower temperature is not adequate for most applications. We develop a new algorithm, BINMATNEST (binary matrix nestedness temperature calculator), that overcomes these difficulties.

Methods BINMATNEST implements a line of perfect order that is uniquely defined, uses genetic algorithms to determine the reordering of rows and columns that leads to minimum matrix temperature, and provides three alternative null models to calculate the statistical significance of matrix temperature.

Results The NTC performs poorly when the input matrix has checkerboard patterns. The more efficient packing of BINMATNEST translates into matrix temperatures that are lower than those computed with the NTC. The null model implemented in the NTC is associated with a large frequency of type I error, while the other null models implemented in BINMATNEST (null models 2 and 3) are conservative. Overall, null model 3 provides the best performance. The nestedness temperature of a matrix is affected by its size and fill, but the probability that such a temperature is obtained by chance is not. BINMATNEST reorders the input matrix in such a way that, if fragment size/isolation plays a role in determining community structure, there will be a significant rank correlation between the size/isolation of the fragments and the way that they are ordered in the packed matrix.

Main conclusions The nestedness temperature of presence–absence matrices should not be calculated with the NTC. The algorithm implemented by BINMATNEST is more robust, allowing for across-study comparisons of the extent to which the nestedness of communities departs from randomness. The sequence in which BINMATNEST reorders habitat fragments provides information about the causal role of immigration and extinction in shaping the community under study.

Keywords
BINMATNEST, conservation, disorder, extinction, habitat fragmentation, immigration, nestedness index.

www.blackwellpublishing.com/jbi
doi:10.1111/j.1365-2699.2006.01444.x

INTRODUCTION

One of the main goals of community ecology is the identification and explanation of non-random patterns of species composition. One of the patterns that has been identified is the nestedness of insular biotas: isolated communities often form nested subsets, with species on species-poor islands constituting subsets of those present on richer islands. The nested pattern was first proposed by Darlington (1957), and has proved to be a very common one (Simberloff & Martin, 1991).

Several ecological processes can produce a nested pattern in a community. Differential dispersal can lead to nestedness, if accessible 'islands' are colonized by large numbers of 'continental' species and inaccessible islands are only colonized by species with high dispersal capabilities (Darlington, 1957). Alternatively, nestedness may be the result of differential extinction, if species in the colonization pool are equally likely to reach all fragments but some species can only persist in, say, the larger fragments (Brown, 1971; Patterson, 1980). The two processes are not mutually exclusive, and some nested patterns are thought to be simultaneously affected by differential colonization and extinction (Lomolino, 1996).

When studying the nestedness of the biotas present in a set of islands, mountain tops or otherwise isolated fragments, there are three basic problems: (1) how to calculate the nestedness of the community, (2) how to determine the probability that chance alone is responsible for the observed pattern, and (3) how to interpret deviations from random expectations. Several indexes can be used to calculate the nestedness of a community (reviewed by Wright *et al.*, 1998). A very popular one, developed by Atmar and Patterson (Atmar & Patterson, 1993) and cited 147 times (ISI database search, August 2005), is the 'nestedness temperature', T. In fact, the use of T to study the nestedness of systems such as plant–pollinator networks (Bascompte *et al.*, 2003) and host–parasite interactions (Norton *et al.*, 2004; Zelmer *et al.*, 2004), for which the index was not originally developed, is becoming increasingly common.

Calculating the nestedness temperature of a community is no trivial task, and most researchers analyse their data using the 'Nestedness Temperature Calculator Program', or NTC (Atmar & Patterson, 1995), freely available on the web (http://www.aics-research.com/nestedness/tempcalc.html). Fischer & Lindenmayer (2002) have pointed out that the NTC is often used as a black box, and that this usage may lead researchers to incorrect conclusions. Although Fischer & Lindenmayer (2002) are concerned with the null model used to ascertain the statistical significance of a matrix temperature, the problem goes further than this: the definition provided by Atmar & Patterson (1993) is consistent with many different algorithms, and it does not specify a one-to-one relationship between a matrix and its temperature. There being (to our knowledge) no published account of the algorithm implemented by the NTC, it is impossible to determine which of the many nestedness temperatures compatible with the definition of Atmar and Patterson (Atmar & Patterson, 1993) the software is calculating.

The aims of this paper are: (1) to examine the definition of nestedness temperature proposed by Atmar and Patterson (Atmar & Patterson, 1993), highlighting its ambiguities, (2) to describe the algorithm implemented by the NTC, pointing out the aspects of this algorithm that are at odds with the original definition of nestedness temperature, (3) to develop a new algorithm that avoids these complications, and (4) to study how the nestedness temperature can be used to infer the ecological processes responsible for community assemblage. Point 2 is based on the Visual Basic source code of the NTC, kindly provided by W. Atmar. Point 4 develops suggestions pioneered by Kadmon (1995) and Lomolino (1996) about how to ascertain the extent to which differential immigration and differential extinction affect community structure.

Although this paper provides sufficient information for the implementation of the algorithm we propose, users can see the BINMATNEST (binary matrix nestedness temperature calculator) program as currently implemented in Appendices S1 and S2 of the Supplementary Material. The program may also be viewed on the internet (at http://www.eeza.csic.es), and the C++ code is available from the corresponding author upon request.

THE NESTEDNESS TEMPERATURE OF A MATRIX

Before going any further, it will be convenient to introduce some notation. The list of species present in a series of islands, fragments or habitat types can be summarized in a presence–absence matrix, M. In this matrix, the element corresponding to row i and column j is set to $M_{ij} = 1$ if species j is present in fragment (or island) i, and to $M_{ij} = 0$ otherwise. We will say that the matrix M has a 'presence' or a 'one' in position (i, j) if $M_{ij} = 1$, and that it has an 'absence' or a 'zero' if $M_{ij} = 0$. The 'fill' of a matrix, ϕ, is defined as the proportion of its cells that denote the presence of a species at a site. Finally, there exists a correspondence that associates every $n \times m$ matrix (n rows, m columns) with the unit square in the (x, y)-plane, $[0, 1] \times [0, 1]$, in such a way that to the element M_{ij} corresponds the point $((j - 0.5)/m,(n - i + 0.5)/n)$. Thus, M_{11} is associated with the top-left corner of the square, and M_{nm} with its bottom-right corner. (The transformation of a rectangle into a square preserves the ratio between distances along a line and poses no problem for calculating the temperature of the matrix.)

Calculating the temperature of a binary matrix involves three steps (Atmar & Patterson, 1993). We first compute an isocline of perfect order, which is a curve that, in a perfectly nested matrix of the same size and fill, would separate cells denoting the presence of a species at a site from cells denoting its absence. We then reorganize the matrix, permuting rows and columns in the way that maximizes its nestedness. In the final step, we associate with each absence above the isocline and with each presence below it a normalized measure of distance to the isocline. The temperature of the matrix is the sum of these distances, normalized in such a way that it ranges between 0 for a perfectly nested matrix and 100 for a maximally 'unnested' matrix. Traditionally, matrix

temperature is given in 'degrees' (Atmar & Patterson, 1993). However, matrix temperature is a percentage, and as such is an non-dimensional quantity, and we will therefore avoid giving any units to it.

The isocline of perfect order is 'a line of smoothest transition' drawn from $(0.5/m,\ 0.5/n)$ to $((m - 0.5)/m,(n - 0.5)/n)$, and such that the area of the unit square above the line exactly equals the fill of the matrix, ϕ (Atmar & Patterson, 1993). This definition is, however, ambiguous. To start with, given two continuous functions, f and g, and given that the derivatives f^n and g^n exist for all n, we know of no way to decide which function is 'smoother'. We take 'a line of smoothest transition' to imply a strictly monotonic function [i.e. a function f such that $f(x) > f(x')$ if and only if $x > x'$], with no inflection point [i.e. always concave or always convex in the interval (0, 1): this seems to be implied by the assertion that the isocline of perfect order will be concave or convex depending on whether the fill of the matrix is smaller or greater than 0.5 (Atmar & Patterson, 1993)], and with no singularities in (0, 1). Even with these restrictive specifications, the isocline of perfect order is not uniquely defined: infinitely many functions, for instance a polynomial or an exponential function, could be found satisfying all the requirements. The ambiguity in the definition of the isocline of perfect order is intrinsic, and essentially it derives from the fact that two perfectly nested matrices of the same dimension and fill need not be equal.

Let us assume, for the time being, that we can actually calculate the isocline of perfect order. Then, a matrix element M_{ij} is 'unexpected' if it denotes a presence below the isocline or an absence above it. The unexpectedness u_{ij} of such an element depends on how far from the isocline it lies. To be precise,

$$u_{ij} = \left(\frac{d_{ij}}{D_{ij}}\right)^2, \tag{1}$$

where d_{ij} is the distance from the point A representing the cell in question, $A = ((j - 0.5)/m, (n - i + 0.5)/n)$, to the isocline along a line with slope -1 (distance between points A and B in Fig. 1), and D_{ij} is the length of the segment with slope -1, running through point A, contained within the unit square (distance between E and F in Fig. 1). The total unexpectedness of the matrix is

$$U = \frac{1}{n \cdot m}\sum u_{ij}, \tag{2}$$

where the sum is conducted over all unexpected cells, and the temperature of the matrix is

$$T = kU, \tag{3}$$

where $k = 100/U_{\max}$ and $U_{\max}$ is the maximum value that U can achieve for any conceivable matrix, $U_{\max} \sim 0.04145$ (Atmar & Patterson, 1993).

Clearly, the value of T thus computed will change if we permute the columns or rows of a matrix, although the information contained in the matrix is not affected by such permutations. The temperature of the matrix is therefore defined as the minimum value of T that can be obtained by permuting rows and columns (Atmar & Patterson, 1993).

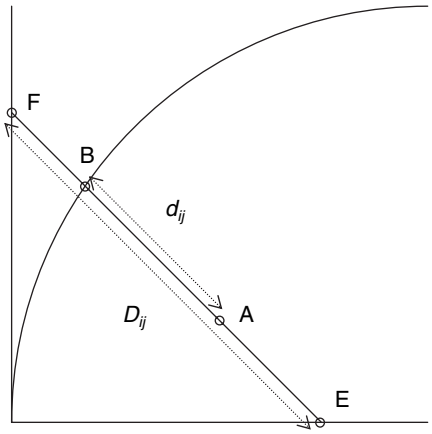

Figure 1 Representation of the elements involved in the computation of unexpectedness, $u_{ij} = (d_{ij}/D_{ij})^2$. The isocline of perfect order divides the unit square into two regions. The distance from a matrix element to the isocline, d_{ij}, is the distance between point A representing the matrix element and point B, itself the intersection between the isocline and a straight line with slope –1 running through A. D_{ij} is the length of the segment with slope –1, running through A, contained within the unit square (distance between E and F).

Once the temperature of a matrix is known, it may be useful to know whether a matrix with this temperature is likely to be obtained by chance. There is no analytical method to do this, and we must resort to Monte Carlo simulations.

The following three sections look in more detail at the isocline of perfect order, how the presence–absence matrix is packed, and how deviations from expected values are calculated. Each section compares the algorithms implemented by the NTC and BINMATNEST. The discussion of the algorithms implemented by the NTC is based on the analysis of the program source code.

THE ISOCLINE OF PERFECT ORDER

The NTC implements the isocline of perfect order in parametric form, $(x(t),\ y(t))$, where both $x(t)$ and $y(t)$ are cubic polynomials of the parameter t. Strictly speaking, the resulting isocline need not be monotonic, it can have inflection points, and it is not symmetric to reflections over the $x + y = 1$ diagonal. In practice, however, the isocline is almost always well behaved.

The NTC defines the isocline of perfect order as a function of a parameter p, $(x(t, p), y(t, p))$, and adjusts the value of p in such a way that the number of cells denoting absences above the line exactly equals the number of cells denoting presences below it. [A matrix element, M_{ij}, is said to be above the isocline of perfect order if the point $((j - 0.5)/m,(n - i + 0.5)/n)$ is above the isocline.] Notice that this is not exactly the same

thing as adjusting the parameter p until the area above the curve equals the fill of the matrix, ϕ. In particular, the number of presences below the curve and the number of absences above it are integer variables, and hence discrete. This implies that, in general, there is not a single value of p, but a continuous range of values (p_1, p_2), such that $(x(t, p), y(t, p))$ divides the unit square into two regions with as many absences in the upper region as presences in the bottom one whenever $p_1 < p < p_2$.

As explained in the previous section, there is no a priori way to determine the 'best' isocline of perfect order, and it would be perfectly possible to use the same functional form as chosen for the NTC, adjusting the value of p in such a way that the area above the isocline equals the fill of the matrix. However, a parametric function like the one used by the NTC significantly complicates the numerical algorithms and slows down calculations. For this reason, BINMATNEST defines the isocline of perfect order as

$$f(x;p) = \frac{0.5}{n} + \frac{n-1}{n} \cdot \left(1 - \left(1 - \frac{m \cdot x - 0.5}{m-1}\right)^{p}\right)^{1/p}. \quad (4)$$

This function is monotonically increasing in x and has no singularities in the interval $0 < x < 1$. Whatever the fill of our presence–absence matrix, there is $p \in (0,\infty)$ such that the area above the curve equals ϕ. As Fig. 2 shows, once the parameters are properly adjusted there is virtually no difference in the isoclines of perfect order used by BINMATNEST and by the NTC.

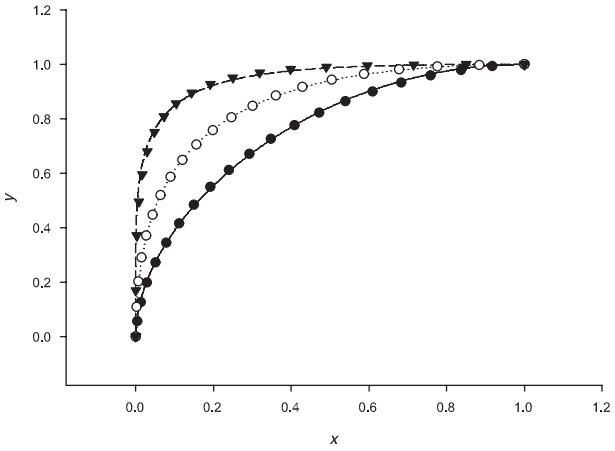

Figure 2 The isocline of perfect order as calculated with the nestedness temperature calculator (symbols) and BINMATNEST (lines) for various matrix fillings: $\phi = 0.05$ (triangles), 0.15 (open circles) and 0.25 (closed circles). The x and y coordinates correspond to the j and i indexes of the matrix elements (respectively) after the matrix is transformed into the unit square. For most parameter values, there is essentially no difference between the two methods for calculating the isocline. The function defined by equation 4 simplifies calculations, but leads to virtually identical results to the function used by the nestedness temperature calculator.

PACKING THE PRESENCE–ABSENCE MATRIX

For any but the smallest matrices, finding the permutation of rows and columns that leads to minimum temperature is a very difficult task. Essentially, the problem is that the number of possible arrangements ($n!\ m!$ in the absence of duplicated rows and columns) is so large that one cannot compute the temperature for every possible configuration and select the configuration leading to the minimum temperature. One can try to solve the problem using some sort of directional minimization scheme, but the abundance of local minima makes it very difficult to attain the global minimum.

The NTC does not really attempt to solve this problem. Instead, it follows a simple heuristic to order rows and column in such a way as to approach the optimal packing. To understand how this is done, consider first the ordering of the columns. For each column j, we can define scores $s(j)$ and $t(j)$ as follows:

$$\left.\begin{aligned} s(j) &= \sum_{i|M_{ij}=1} i^2 \\ t(j) &= \sum_{i|M_{ij}=0} (n-i+1)^2 \end{aligned}\right\} \quad (5)$$

We now find two permutations of the integers 1, 2,…, m, $\{l_1, l_2, \ldots, l_m\}$ and $\{k_1, k_2, \ldots, k_m\}$, such that $t(l_i) \leq t(l_{i+1})$ and $s(k_i) \geq s(k_{i+1})$. In other words, the l sequence arranges columns in ascending order of their t score (packing by absences), and the k sequence arranges columns in descending order of their s score (packing by presences). Although there need not be a perfect correlation between ordering by t (or s) score and minimizing temperature, for the value of T depends subtly on the exact placement of the isocline of perfect order, both orders make some sense a priori. The NTC takes into account both scores, arranging columns in a sequence that is a compromise between the l and k sequences. First it arranges the matrix by columns, then by rows, and the entire process is iterated eight times. (The order is reversed if there are more rows than columns.)

After rearranging the matrix, the NTC removes empty rows and columns and duplicated all-presence rows and columns. That is, if there are i rows (or columns) without empty cells, $i - 1$ of them are removed. Considering its simplicity, the algorithm works very well. It can, however, easily become trapped into cycles that produce suboptimal packing. For instance, if we use a 10×10 diagonal matrix as input for the NTC, we obtain very different results depending on how we order rows and columns in the input matrix: with the standard diagonal matrix ($M_{ij} = \delta_{ij}$), the output of the NTC is $T = 52.64$, while if we permute rows (or columns) in such a way that the presences are arranged from the bottom-left to the upper-right corners, the output is $T = 29.92$. (Both matrices contain the same information, and it is a trivial task to go from one to the other by permuting rows or columns.) This is because the packing algorithm actually leaves the input matrices untouched. Although for most input matrices the packing algorithm

performs reasonably well, the same problem appears in most large matrices at a local scale (i.e. affecting a small number of rows and columns). In particular, if the species present at two sites are the same, except for two species, one of which is present at one site while the other is present at the other site, the packing algorithm will give different results depending on which site is associated with which matrix row. That is, the packing algorithm of the NTC has difficulties dealing with 'checkerboard' patterns.

In order to improve the algorithm implemented by the NTC, we first note that the question of how to organize rows and columns to minimize the temperature of a matrix is not unlike the travelling salesman problem (TSP). Given a set of geographic locations, the TSP consists in searching for the shortest possible path traversing every location (Christofides, 1976; Lawler *et al.*, 1985). After decades of intense research (for a recent review, see Gutin & Punnen, 2002), the TSP still lacks a definite solution. It would therefore be ludicrous to search for the ultimate solution of the matrix-packing problem. We can, nevertheless, search for inspiration in the abundant TSP literature. Researchers working on the TSP have used two main approaches: linear programming (as in Gamarnik *et al.*, 2004) and genetic algorithms (GA, as in Soak & Ahn, 2004). Adapting the complex linear-programming solutions of the TSP to packing the matrix is well beyond our capabilities, but the simplicity of GA makes them more malleable. We have therefore opted to use a GA to pack the matrix.

A GA operates by producing an initial set of *PopSize* possible solutions that are improved by the production of new variants with selection of the best-performing ones. In BINMATNEST, selection takes place as follows. In every generation, *TourSize* individuals are selected at random from the entire population and the worst two individuals (the ones with the highest temperature) are replaced by 'offspring' of the two best-performing individuals of the group. This process is iterated for *nbGen* generations, and the best-performing solution is used to calculate the temperature of the input matrix. Note that the algorithm calculates the actual temperature of each proposed solution, instead of trying to minimize some other metric that is assumed to be correlated with the temperature. We now explain how matrix temperature is calculated, how offspring are produced, and how the initial population is created.

Before entering the GA, BINMATNEST orders columns (left to right) and rows (top to bottom) in descending order according to the number of presence-denoting cells they have, and it removes empty and duplicated all-presence rows and columns. It then calculates the fill of the matrix (ϕ = number of presence-denoting cells divided by the product $n \times m$) and the value of p (equation 4) corresponding to such fill. This determines the isocline of perfect order, from which we calculate the unexpectedness matrix, U, that has for elements the u_{ij} of equation 1. This matrix is calculated only once. Now, given a proposed solution (that is, a reordering or rows and columns), the computation of its temperature is very fast, because it simply involves adding up the elements of the U matrix corresponding to 'unexpected' cells (presence-denoting cells below the isocline, absence-denoting cells above it) of the proposed solution, and a final multiplication (equations 2 and 3).

A proposed solution is coded by two 'chromosomes': one for the rows and one for the columns. The row chromosome contains a permutation $\{r_1, \ldots, r_n\}$ of the numbers $1, \ldots, n$, and the column chromosome a permutation $\{c_1, \ldots, c_m\}$ of the numbers $1, \ldots, m$. Consider the row chromosome: r_i indicates the position that the ith row of the original matrix should take in the proposed solution. When we want to produce an offspring from a well-performing solution, we first select a second parent at random from the entire population. Let $\{w_1, \ldots, w_n\}$ and $\{p_1, \ldots, p_n\}$ be the row chromosomes of the well-performing solution and its partner, respectively. The row chromosome of the offspring, $\{o_1, \ldots, o_n\}$, equals $\{w_1, \ldots, w_n\}$ with probability 0.5, and it is otherwise produced by combining the information from both parents, using the following algorithm.

1. Select an integer k, $1 \leq k < n$, at random (uniform distribution).
2. Let $o_i = w_i$ for $i = 1, \ldots, k$.
3. For $i = k + 1, \ldots, n$, $o_i = p_i$ if $p_i \notin \{w_1, \ldots, w_k\}$.
4. For those i for which $i > k$ and $p_i \in \{w_1, \ldots, w_k\}$, the value of o_i is chosen at random from all positions not yet used.

The same procedure applies to the column chromosome. The choice of combining information from both parents or not is made independently for each chromosome, subject to the condition that at least one chromosome must combine information from both proposed solutions. For all the results shown in this paper, random numbers with uniform distributions were generated with the 'ran1' function of Press *et al.* (1992), initialized with the computer clock.

As a final step, each offspring chromosome suffers a 'mutation' with probability 0.1. For row chromosomes, mutations consist in choosing random integers k_1 and k_2 in $[1, n]$ and producing a cyclic permutation of the elements $r_{k1}, \ldots, r_{k2}$.

The starting population is created as follows: the original matrix (after removing empty and duplicated all-presence rows and columns) constitutes the first proposed solution. The second proposed solution is the one obtained with the packing algorithm of the NTC. Solutions 3 to 13 are produced with an algorithm not unlike the one used by the NTC packing routine. Assuming that there are more columns than rows, we first sort columns, then rows, and iterate the process four times (the order is reversed if there are more rows than columns). To sort columns, we calculate a score for each column, $z(j) = x - s(j) + (1 - x)t(j)$, where $s(j)$ and $t(j)$ are given by equation 5 and x takes values 0, 0.1, … and 1 for proposed solutions 3, 4, … and 13, respectively. Columns are ordered in descending order of their z scores. (An analogous procedure is used for sorting rows.) All remaining proposed solutions of the starting population are obtained as random 'mutations' of these 13.

CHOICE OF NULL MODEL

More often than not, we want to know the probability that the nestedness of a 'random' matrix is higher than (or equal to) the nestedness of our data; that is, the probability that the temperature of a random matrix is lower than or equal to the temperature of our system. Unfortunately, there is no a priori way to specify what a random matrix is, and researchers will have to be very careful when evaluating the statistical significance of their results. The NTC creates a number of 'random' matrices by letting each individual element be equal to 1 with probability ϕ (the fill of the input matrix). It calculates the temperature of each random matrix and returns the proportion of matrices with temperature lower than or equal to the temperature of the input matrix. This will not always be an appropriate null model (Fischer & Lindenmayer, 2002).

To solve this difficulty, BINMATNEST returns three '*p*-values', associated with different null models. The first one is calculated using the same null model as implemented in the NTC (null model 1). Null model 2 is the one proposed by Fischer & Lindenmayer (2002). For each column, it calculates the fraction of presence-denoting cells,

$$\rho_j = \frac{1}{n}\sum_{i=1}^{n} M_{ij}, \qquad (6)$$

and the random matrices, A, are created by letting $A_{ij} = 1$ with probability ρ_j (regardless of the value of i). Some authors have argued that the null model should control for the number of ones in each row and column (Connor & Simberloff, 1979; Sanderson *et al.*, 1998; Brualdi & Sanderson, 1999). Although these authors certainly have a point, there is as yet no known efficient algorithm that can produce random elements of the corresponding null space in such a way that every element of the null space has the same probability of being generated. Several published algorithms will generate random matrices satisfying the row and column constraints (Sanderson *et al.*, 1998; Gotelli & Entsminger, 2001), and some of these algorithms are actually very efficient (Miklós & Podani, 2004), but they do not sample the null space uniformly (Miklós & Podani, 2004). Likewise, Miklós & Podani (2004) develop an algorithm that produces a distribution of random matrices that is asymptotically uniform, but in practice the algorithm is too slow for all but the smallest matrices. It has been suggested that combining a fast, biased algorithm with a few steps of the asymptotically uniform algorithm might do the trick (Miklós & Podani, 2004), but it does not (M.A. Rodríguez-Gironés, unpubl. results). For this reason, null model 3 of BINMATNEST does not attempt to control exactly for the row and column totals. Instead, it follows a strategy introduced by Bascompte *et al.* (2003) to calculate the *p*-value. The fraction of presence-denoting cells is computed for rows, ρ_{ri}, and columns, ρ_{cj}, and for every random matrix A the element A_{ij} is set equal to 1 with probability $(\rho_{ri} + \rho_{cj})/2$ (Bascompte *et al.*, 2003). BINMATNEST allows users to decide how many random matrices are generated to calculate *p*-values. The user is also free to select the values of *PopSize*, *TourSize* and *nbGen* that are most suitable for the problem under consideration. All the results shown here correspond to *PopSize* = 35, *TourSize* = 7 and *nbGen* = 2000. Increasing these values will improve the accuracy of the results, but at the price of increasing computing time.

It seems to us that there is no a priori way to determine which (if any) of the null models implemented by BINMATNEST is the 'correct' one, because the right null model to use will depend on the constraints to which the presence–absence matrices are subject. These constraints need not be universal, and it would make little difference if they were, because we ignore them. We can, however, answer a more modest question. Namely, given the temperature of a presence–absence matrix, if the matrix was actually subject to certain constraints, and we calculated the probability that a random matrix had a lower temperature using a sample space subject to a different set of constraints, what errors would we introduce?

To answer this question, we generate a 'template' matrix as follows. We select integers n and m at random (independent uniform distributions in the range 10–30) and a real number P (uniform distribution, range 0.2–0.8) and let $T_{ij} = 1$ with probability P, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$. From this template matrix, we generate three 'data' matrices using the algorithms explained above in the description of null models 1–3. We finally compute, for each data matrix, the *p*-value associated with each null model. The process was iterated 500 times. Because the original data matrices were themselves random, the *p*-values should be uniformly distributed between 0 and 1, and a cumulative plot of the *p*-values obtained should follow a straight line with slope 1. As Fig. 3 shows, this was never the case. All null models provide conservative tests (cumulative plot below diagonal for small *p*-values) when the data matrix was generated with the same constraints as assumed in the calculation of the *p*-value. Using null model 1 leads to a high incidence of type I errors (cumulative plot above diagonal for small *p*-values) when the data matrix is subject to a different set of constraints, while using null models 2 and 3 always leads to conservative tests and a high incidence of type II errors (with the exception of null model 3, which performs just right when data matrices are generated with null model 2 – the 'type I errors' that it introduces for large *p*-values are not a practical concern). Overall, null model 3 seems to perform better (smaller type I error) than the others, and the use of null models 1 and 2 would only seem to be justified if there is evidence supporting the claim that the data under consideration are subject to those specific constraints.

At first sight, it might seem odd that the diagonal plots in Fig. 3 deviate from the straight line, since in these plots the data matrices were generated with the same algorithm as used to calculate the *p*-values. The point is that the parameters of the algorithm were different. Consider, for example, the upper-left plot (null model 1). We first create a template matrix. If the

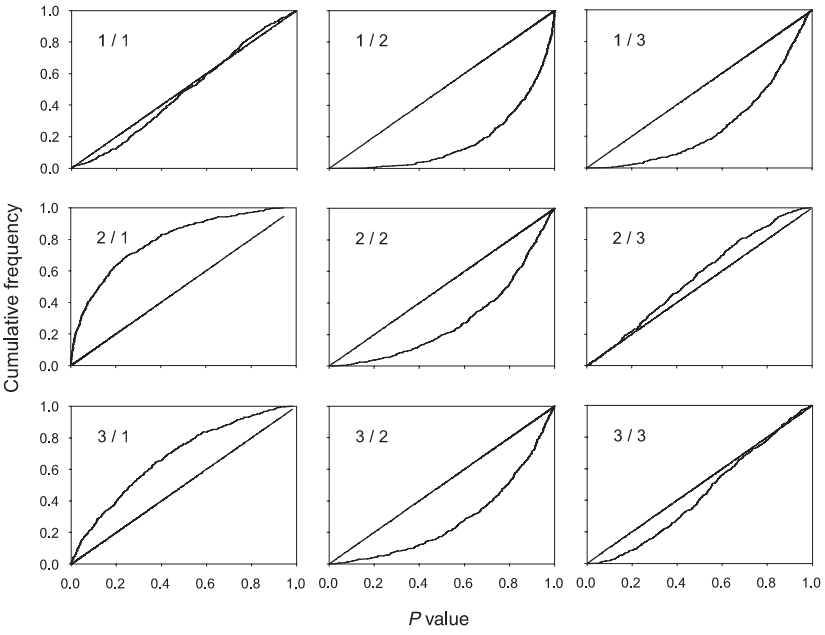

Figure 3 Cumulative frequency of the *p*-values associated with the nestedness temperature of random matrices. The label *i*/*j* in each plot indicates the null model used to generate data matrices (*i*) and to calculate their *p*-values (*j*). Straight lines have slope 1. Curves above the straight line represent a high tendency for the test to produce type I errors, while curves below the straight line represent conservative tests (high incidence of type II errors).

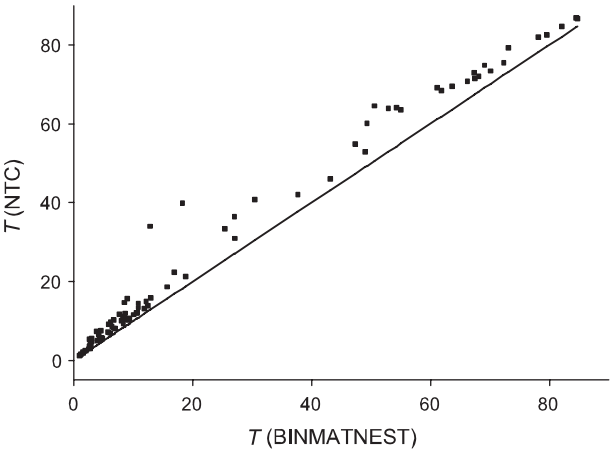

Figure 4 Relationship between the temperature values obtained with BINMATNEST (abscissa) and the nestedness temperature calculator (ordinates) for 90 computer-generated matrices. The straight line represents the line of equality, and points represent individual matrices. All points lie above the diagonal.

fill of the template matrix is ϕ, the elements of the 'data' matrix will be set equal to 1 with probability ϕ. As a result, the expected fill of the data matrix is itself ϕ, but in most particular trials, as a result of sampling error, the actual fill will be $\phi' \neq \phi$. It is ϕ' that will be used to generate the matrices from which the *p*-value is calculated – hence the discrepancy.

COMPARISON OF THE TWO ALGORITHMS

Figure 4 shows the relationship between the temperatures calculated with the two algorithms for a set of 90 computer-generated matrices. The value obtained with BINMATNEST was always lower than the value obtained with the NTC, the difference being occasionally quite large. When the NTC managed to pack the input matrix efficiently, there was hardly any difference between the two algorithms, but when the packing algorithm of the NTC becomes trapped in a cycle it can lead to a serious overestimation of the matrix temperature. Although the packing algorithm implemented in BINMATNEST does not necessarily find the optimal packing either, it is more robust and less likely to become trapped in local minima than the algorithm used by the NTC. As a particular example, BINMATNEST finds the same temperature ($T = 30.28$) for the 10×10 diagonal matrix, regardless of how the ones of the input matrix are arranged. Overall, there was a good correlation between the temperatures found by both algorithms ($R^2 = 0.98$), but the slope of the regression was slightly greater than one (1.05; 95% confidence limits: 1.02–1.08) and the intercept was greater than zero (2.39; 95% confidence limits: 1.36–3.43).

Repeating the exercise with real data gives intriguing results (Fig. 5). We analysed two data sets. The first one consists of 37 of the absence–presence matrices that come with the NTC package (the first 37 with less than 250 species, in alphabetical order). The second one consists of 37 mutualistic pollination networks, mostly those reported by Bascompte *et al.* (2003) and Olesen & Jordano (2002), but including five matrices published elsewhere (Kato, 2000 Yamazaki & Kato, 2003; Dupont *et al.*, 2003; Stald *et al.*, 2003) and three unpublished matrices kindly provided by J. M. Olesen.

The relationship between the temperature values obtained with the TNC and BINMATNEST differ significantly for the two data sets ($F(1,70) = 10.75$, $p < 0.0016$ for a test of the homogeneity of slopes based on a GLIM analysis). While the data for the presence–absence matrices fall along the regression line obtained from the computer-generated matrices ($T_{NTC} = 1.82 + 1.22T_B$, where T_{NTC} and T_B are the

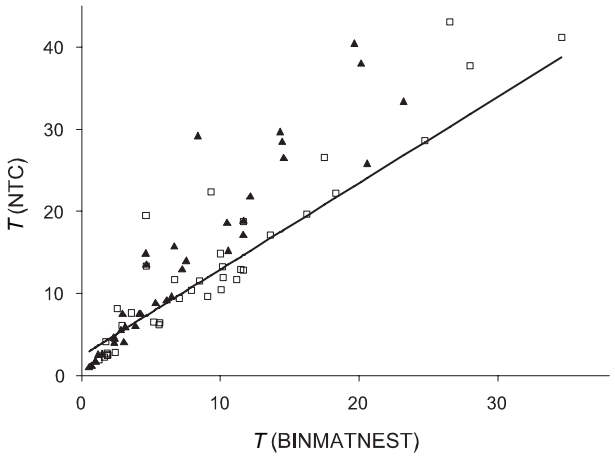

Figure 5 Relationship between the temperature values obtained with BINMATNEST (abscissa) and the nestedness temperature calculator (ordinates) for: presence–absence matrices representing island biotas (empty squares); and interaction matrices representing pollination networks (black triangles). The straight line represents the regression line for the computer-generated matrices of Fig. 4.

temperature values obtained with the TNC and BINMATNEST, respectively; $n = 37$, $R^2 = 0.87$), the data for the pollination networks do not: the NTC performs poorly on these data, grossly overestimating matrix temperature ($T_{NTC} = 1.32 + 1.65T_B$; $n = 37$, $R^2 = 0.88$). In both cases, the slope of the relationship differed significantly from 1 (95% confident intervals were 1.44–1.86 for pollination networks and 1.07–1.38 for presence–absence data). It could be argued that the nestedness temperature was not intended for use with pollination networks, but this is not the point: what is important here is that the error introduced by the packing algorithm of the NTC depends on the structure of the data. Incidentally, the analysis also shows that there are differences in the way that pollination networks and island biotas are assembled, but this result, in itself, is hardly surprising, and the nature of these differences remains at present elusive.

INTERPRETING NESTEDNESS

Whichever way we have calculated the nestedness temperature of our presence–absence matrix and whatever the extent that it deviates from randomness, we may well wonder what it all means. In this section, we deal with two issues related to the interpretation of nestedness temperature: across-study comparisons and causality.

External validity

If the nestedness temperature of butterflies in archipelago A is $T_A = 23$, and in archipelago B it is $T_B = 37$, can we conclude that the subsets of butterflies in A islands are more nested than those in B? The nestedness temperature is normalized in such a way that it will always be a non-negative number smaller than

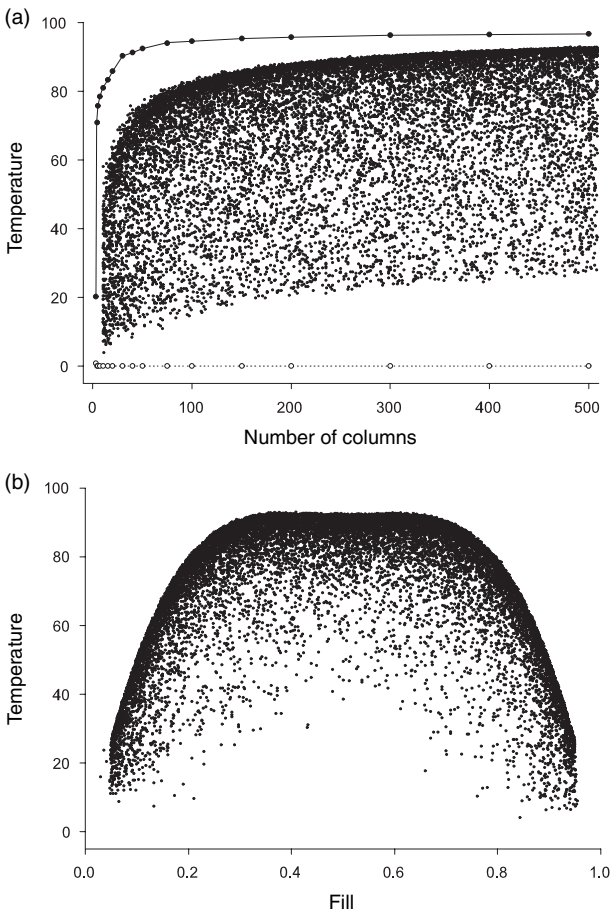

Figure 6 Dependence of temperature on (a) matrix size and (b) fill for random square matrices. Each point represents one matrix. In the top panel, circles linked with lines represent the maximum and minimum temperatures that can be obtained for each matrix size.

or equal to 100. This, however, does not imply that there is no relationship between a matrix structure (specifically, its size and fill) and its temperature.

To study the relationship between matrix structure and temperature, we first generated 15,000 square matrices by choosing at random their dimension (uniform distribution, range 10–500) and a real number P (uniform distribution, range 0.05–0.95). Each element of the matrix was then set equal to 1 with probability P, and the nestedness temperature of each matrix was calculated. As Fig. 6 shows, the distribution of matrix temperatures changed with the size and fill of the matrices. For every matrix size or fill, the nestedness temperatures of the matrices attain a broad range of values, with most temperatures concentrated around the upper limit of this range. Both the upper and lower limits of this range show definite patterns, so that temperature increases with matrix size and attains its maximum value for intermediate fills (Fig. 6). The temperature values obtained with these random matrices do not, however, span the entire range of possible temperatures. It is possible to search specifically for matrices with very high (or low) temperatures. Figure 6 also shows the relationship between

matrix size and the highest and lowest temperature values that can be reached. Whatever the size of the matrix, it is possible to produce nested matrices with temperature very close to zero. The maximum temperature that can be reached, however, is an increasing function of matrix size. Although the value shown in Fig. 6 may underestimate maximum temperature for large matrices, the value obtained for small matrices was obtained with an exhaustive search of all possible matrices and can therefore be taken at face value.

What does not depend on the fill and size of a matrix is the probability that its degree of nestedness is obtained by chance. We generated 10 random square matrices for each combination of size (25, 75, 125, … and 475 rows and columns) and fill (0.1, 0.3, …, 0.9), and calculated their temperature and their associated *p*-value (null model 3). The *p*-values (or rather, the arcsines of their square roots – the transformation improved the normality of residuals and homogeneity of variances) did not change significantly either with the size of the matrix ($F = 0.882$, 9 d.f., $p = 0.54$), its fill ($F = 1.939$, 4 d.f., $p = 0.10$), or their interaction ($F = 1.182$, 36 d.f., $p = 0.22$).

It follows from the above that a direct comparison of the temperatures of two systems is meaningless unless the matrices representing them have the same size and fill, whereas a comparison of their *p*-values is legitimate.

Differential immigration or extinction?

Some authors have attempted to use the concept of nestedness to ascertain whether selective immigration, selective extinction, or both, are responsible for the structure of particular communities (Kadmon, 1995; Lomolino, 1996). Thus, Lomolino (1996) produced, for each 'archipelago' under consideration, two presence–absence matrices. In one of them islands were ordered according to their size, and in the other by their isolation from the mainland. He then computed the nestedness of each matrix (using an index that does not involve re-sorting rows and columns) and essentially derived a multivariate test that allowed him to study the effects of each variable (size and isolation) on nestedness, after partialling out the effects of the other. If immigration and extinction are correlated with isolation and size of the islands, the test gives information about the extent to which selective immigration and extinction are responsible for community structure [see Lomolino (1996) for details, discussion and caveats].

Computing the nestedness temperature of a binary matrix involves re-ordering rows and columns, and therefore Lomolino's (1996) approach cannot be used. Nevertheless, the order in which islands are sorted by the algorithm may contain equivalent information. To ascertain this possibility, we used the following approach.

Generating hypothetical communities structured by immigration or extinction

Consider first a community structured through immigration. Island j was assigned an isolation factor D_j at random (uniform distribution, range 0–1), and species i a dispersal incidence I_i (uniform distribution, range 0–1). The probability that species i colonized island j was then set equal to $\exp(x)/[\exp(x) + \exp(-x)]$, with $x = \beta(I_i - D_j - x_0)$, where $\beta > 0$. For low values of β, colonization is essentially random. For high values, species i can only colonize island j if $I_i > D_j + x_0$. For every value of β that we used, the value of x_0 was chosen in such a way that the expected fill of the presence–absence matrix was equal to 0.2.

A community generated solely through extinction would be generated in essentially the same way, except that the factor associated with island j would indicate its size, and the factor associated with species i would indicate its incidence (its ability to survive in small islands). A community structured through immigration and extinction can be generated by calculating, for each island–species pair, a probability of initial colonization and a probability of long-term establishment. Each probability would be calculated as stated above, except that the value of β need not be the same for immigration, β_{im}, and extinction, β_{ext}. Thus, a high value of β_{im} and a low value of β_{ext} would imply that immigration has played a more important role than extinction in structuring the community.

Evaluating the importance of immigration and extinction from presence–absence matrices

Once we have generated the presence–absence matrix of a hypothetical community, we use BINMATNEST to find the permutation of rows and columns leading to maximal nestedness and use Spearman's rank correlation to test whether

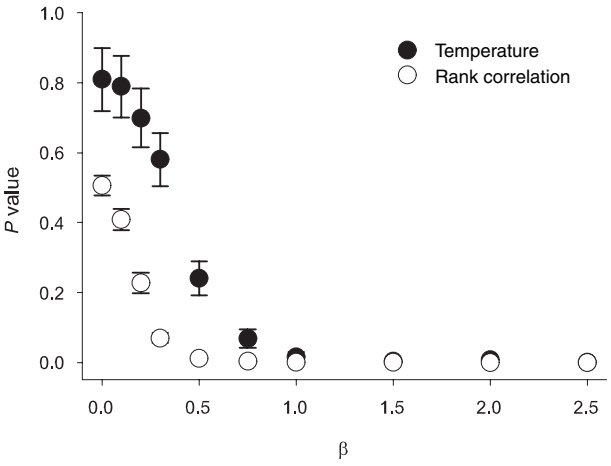

Figure 7 Effect of β on the nestedness temperature and the rank correlation between column order in the packed matrix and island isolation. For each value of β, we generated 100 communities as described in the text. Each community is characterized by the probability that a random matrix (null model 3) has the same or lower temperature (black circles) and by the probability that the correlation between a random permutation of the islands is at least as correlated to their isolation as the sequence in which BINMATNEST orders them for calculating the nestedness temperature of the matrix (empty circles). Symbols represent mean values, and bars represent standard the error of the mean.

islands have been sorted by size or isolation. Figure 7 shows the basic results when the community is structured by a single factor (say, immigration). For each value of β, 100 communities were simulated, each community having a number of islands and species randomly chosen between 20 and 80 (independent uniform distributions). As expected, the *p*-value associated with the nestedness temperature of the matrix and the *p*-value of the rank correlation between island order and isolation decrease when β increases. What is more striking is that the test based on the rank correlation is more sensitive than the test on matrix temperature. For instance, when $\beta = 0.5$, the average *p*-value for the matrix temperature is 0.241 (SE 0.049), while the average *p*-value for the rank correlation is 0.012 (SE 0.005). Notice also that, when $\beta = 0$ (immigration plays absolutely no role in community structure), the *p*-values of the rank correlation have a uniform distribution in the interval (0, 1), while the *p*-values associated with matrix temperature are, on average, greater than 0.5 (i.e. temperature is, in some sense, greater than expected by chance).

The increased sensitivity of the rank correlation test relative to the nestedness temperature test is also apparent when communities are structured through immigration and extinction. Figure 8 shows the cumulative distribution of *p*-values for 100 independent runs with $\beta_{im} = \beta_{ex} = 0.1$. The nestedness temperature of the matrix never deviates from random expectations (at the 0.05 level, null model 3), while 34% of communities had a significant ($p < 0.05$) correlation between island isolation or size and position in the packed matrix.

Table 1 Effects of community size (number of species and islands) and organization rules (value of β_{im} and β_{ex}) on the *p*-value of the rank correlation between island isolation and position in the packed matrix

	SS	d.f.	Mean square	*F*	*p*
Intercept	17.1448	1	17.1448	312.8004	< 0.0001
Number of species	0.5732	1	0.5732	10.4578	0.0012
Number of islands	2.6126	1	2.6126	47.6659	< 0.0001
β_{im}	132.6136	4	33.1534	604.8705	< 0.0001
β_{ex}	9.1471	4	2.2868	41.7213	< 0.0001
$\beta_{im} \times \beta_{ex}$	8.1278	16	0.5080	9.2681	< 0.0001
Error	101.2901	1848	0.0548		

We finally let β_{im} and β_{ex} take the values 0.1, 0.3, 0.75, 1.5 and 2.5 (all 25 possible combinations) and produced 75 independently chosen hypothetical communities for each set. A general linear model analysis, with β_{im} and β_{ex} as factors and number of rows and columns as covariates, shows that all factors have a significant effect on the *p*-value of the rank correlation (after applying the arcsine of the square-root transformation) between island isolation and position in the packed matrix (Table 1). Although, as a result of the large sample size, the effect of community size is very significant, this effect is numerically rather small. The *p*-value of the rank correlation decreases with the number of species and islands (i.e. with sample size) with coefficients −0.001 and −0.002, respectively. The *p*-value of the rank correlation between island isolation and position in the packed matrix is

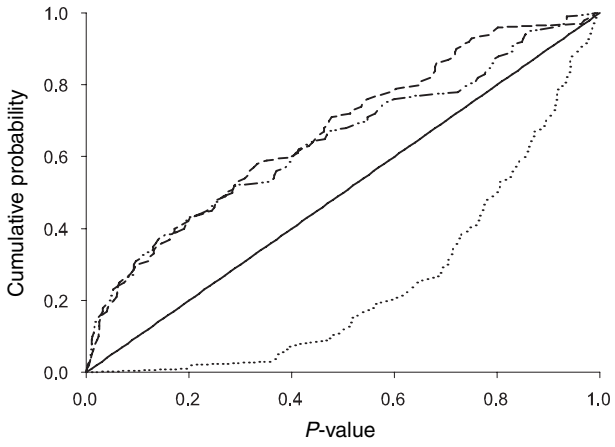

Figure 8 Cumulative distribution of the *p*-values associated with the nestedness temperature (dotted concave line) and rank correlation between island isolation/size and position in the packed matrix (dashed convex lines) for 100 hypothetical communities in which immigration and extinction play a relatively small role ($\beta_{im} = \beta_{ex} = 0.1$). Of the two dashed convex lines, one corresponds to island isolation and the other to island size. Because of the symmetry of the model ($\beta_{im} = \beta_{ex}$ in this particular case) they provide the same information. The solid line is the line of equality, corresponding to the cumulative distribution of a random deviate with uniform distribution.

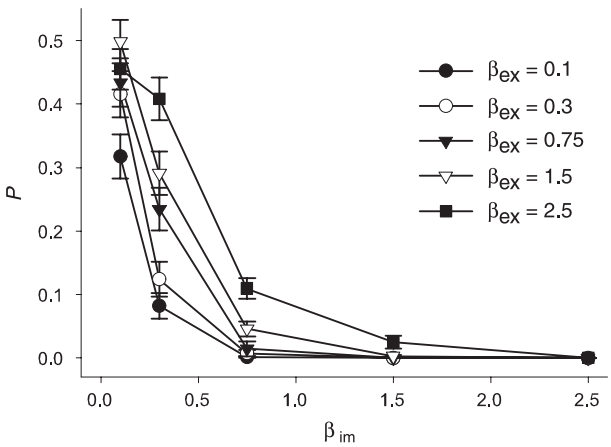

Figure 9 Effect of β_{im} and β_{ex} on the rank correlation between column order in the packed matrix and island isolation. For each combination of β_{im} and β_{ex}, we generated 75 communities as described in the text. Each community is characterized by the probability that the correlation between a random permutation of the islands is at least as correlated to their isolation as the sequence in which BINMATNEST orders them for calculating the nestedness temperature of the matrix. Symbols represent mean values, and bars represent the standard error of the mean. Each set of symbols is associated with a value of β_{ex}.

a decreasing function of β_{im} and an increasing function of β_{ex} (Fig. 9). The correlation thus improves if immigration plays an important role in structuring the community, and deteriorates if the role of extinction becomes stronger. Owing to the symmetry that immigration and extinction play in the model, analogous results are obtained when we analyse the variation of the *p*-value of the correlation between island size and position in the packed matrix.

CONCLUDING REMARKS

In order to calculate the temperature of a binary matrix, one needs to make essentially two choices: one relates to the definition of the isocline of perfect order, and the other to the way the matrix is reorganized before the computations. The choice of the family to which the isocline belongs seems not to be critical, provided that certain basic assumptions are satisfied, but one must make sure that the choice of an isocline from within this family is done in an unambiguous manner. The packing algorithm, on the other hand, can have a strong effect on the results of the computations. Care must be taken that the packing algorithm does not easily get trapped in dynamic cycles or local minima.

The presence of a systematic bias in the results provided by the NTC would pose no special problem, because the difference between the temperatures of two or more matrices would remain meaningful. A completely random error would be undesirable because it would reduce the statistical power of tests, but it would be unlikely to lead researchers to wrong conclusions. What is more worrying is that the noise term introduced by the NTC algorithm depends on the structure of the data. This makes it difficult to compare in any meaningful way results from different studies when the NTC is used to compute the nestedness temperature of a presence–absence matrix. The packing algorithm of BINMATNEST is more robust and therefore less likely to introduce errors that depend on the structure of the data.

Although, by definition, the nestedness temperature of a matrix is a number between 0 and 100 whatever the size and structure of the matrix, the range of values that the matrix temperature can actually take depends on its size and fill. The range of possible temperatures is substantially constrained for small matrices and for matrices with very low or very high fills. It follows that the temperature nestedness of a matrix provides, by itself, relatively little information. On the other hand, the probability that such temperature is obtained by chance can be estimated with a number of null models, and one of them (null model 3) provides reliable information.

When we can estimate the size and isolation of our islands, the sequence in which BINMATNEST reorders islands in the packed matrix can be used to study the relative importance of island size and isolation on structuring the community, and, if size and isolation are directly linked to species immigration and extinction, we can study whether immigration, extinction, or both, are causally responsible for the nested structure of the community.

ACKNOWLEDGEMENTS

Wirt Atmar provided the Visual Basic code of the Nestedness Temperature Calculator Program, and Jens M. Olesen provided unpublished data on pollination networks. We thank Isidro Verdú for his help in writing the input–output sections of BINMATNEST, and two anonymous referees and Jose Alexandre Diniz-Filho for their comments and suggestions. This work was funded by the Spanish Ministry of Science and Technology (project INVASRED, REN2003-06962).

REFERENCES

Atmar, W. & Patterson, B.D. (1993) The measure of order and disorder in the distribution of species in fragmented habitat. *Oecologia*, **96,** 373–382.

Atmar, W. & Patterson, B.D. (1995) *The nestedness temperature calculator, a visual BASIC program, including 294 presence/absence matrices*. AICS Research Inc, University Park, New Mexico, USA and the Field Museum, Chicago, IL, USA.

Bascompte, J., Jordano, P., Melián, C.J. & Olesen, J.M. (2003) The nested assembly of plant–animal mutualistic networks. *Proceedings of the National Academy of Sciences USA*, **100,** 9383–9387.

Brown, J.H. (1971) Mammals on mountaintops: non-equilibrium insular biogeography. *The American Naturalist*, **105,** 467–478.

Brualdi, R.A. & Sanderson, G.J. (1999) Nested species subsets, gaps, and discrepancy. *Oecologia*, **119,** 256–264.

Christofides, N. (1976) *Worst-case analysis of a new heuristic for the travelling salesman problem*. Technical Report, Carnegie Mellon University, Pittsburgh, PA.

Connor, E.F. & Simberloff, D. (1979) The assembly of species communities: chance or competition? *Ecology*, **60,** 1132–1140.

Darlington, P.J. (1957) *Zoogeography: the geographical distribution of animals*. Wiley, New York.

Dupont, Y.L., Hansen, D.M. & Olesen, J.M. (2003) Structure of a plant-flower-visitor network in the high-altitude sub-alpine desert of Tenerife, Canary Islands. *Ecography*, **26,** 301–310.

Fischer, J. & Lindenmayer, D.B. (2002) Treating the nestedness temperature calculator as a ''black box'' can lead to false conclusions. *Oikos*, **99,** 193–199.

Gamarnik, D., Lewenstein, M. & Sviridenko, M. (2004) An improved upper bound for the TSP in cubic 3-edge-connected graphs. *Operations Research Letters*, **33,** 467–474.

Gotelli, N.J. & Entsminger, G.L. (2001) Swap and fill algorithms in null model analysis: rethinking the knight's tour. *Oecologia*, **129,** 281–291.

Gutin, G. & Punnen, A. (2002) *The travelling salesman problem and its variations*. Kluwer Academic Publisher, Dordrecht.

Kadmon, R. (1995) Nested species subsets and geographic isolation: a case study. *Ecology*, **76,** 458–465.

Kato, M. (2000) Anthophilous insect community and plant-pollinator interactions on Amami Islands in the Ryukyu Archipelago, Japan. *Contributions from the Biological Laboratory, Kyoto University*, **29,** 157–252.

Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G. & Shmoys, D.B. (eds) (1985) *The traveling salesman problem*. John Wiley, New York.

Lomolino, M.V. (1996) Investigating causality of nestedness of insular communities: selective immigrations or extinctions? *Journal of Biogeography*, **23,** 699–703.

Miklós, I. & Podani, J. (2004) Randomization of presence–absence matrices: comments and new algorithms. *Ecology*, **85,** 86–92.

Norton, J., Lewis, J.W. & Rollinson, D. (2004) Temporal and spatial patterns of nestedness in eel macroparasite communities. *Parasitology*, **129,** 203–211.

Olesen, J.M. & Jordano, P. (2002) Geographic patterns in plant-pollinator mutualistic networks. *Ecology*, **83,** 2416–2424.

Patterson, B.D. (1980) Montane mammalian biogeography in New Mexico. *Southwestern Naturalist*, **25,** 33–40.

Press, W.H., Teukolsky, S.A., Vettering, W.T. & Flannery, B.P. (1992) *Numerical recipes in C*. Cambridge University Press, Cambridge.

Sanderson, J.G., Moulton, M.P. & Selfridge, R.G. (1998) Null matrices and the analysis of species co-occurrences. *Oecologia*, **116,** 275–283.

Simberloff, D. & Martin, J. (1991) Nestedness of insular avifaunas: simple summary statistics masking complex species patterns. *Ornis Fennica*, **68,** 178–192.

Soak, S.M. & Ahn, B.H. (2004) New genetic crossover operator for the TSP. *Lecture notes in artificial intelligence*, **3070,** 480–485.

Stald, L., Valido, A. & Olesen, J.M. (2003) Struktur og dynamik i rum og tid at et bestøvningsnetværk på Tenerife, De Kanariske Øer. MSc-thesis, University of Aarhus, Denmark.

Wright, D.H., Patterson, B.D., Mikkelson, G.M., Cutler, A. & Atmar, W. (1998) A comparative analysis of nested subset patterns of species composition. *Oecologia*, **113,** 1–20.

Yamazaki, K. & Kato, M. (2003) Flowering phenology and anthophilous insect community in a grassland ecosystem at Mt. Yufu, Western Japan. *Contributions from the Biological Laboratory, Kyoto University*, **29,** 255–318.

Zelmer, D.A., Paredes-Calderón, L., León-Règagnon, V. & Carcía-Prieto, L. (2004) Nestedness in colonization-dominated systems: helminth infracommunities of *Rana vaillanti* Brocchi (Anura: Ranidae) in Los Tuxtlas, Veracruz, Mexico. *Journal of Parasitology*, **90,** 705–710.

SUPPLEMENTARY MATERIAL

The following supplementary material is available for this article online from http://www.Blackwell-Synergy.com

Appendix S1 BINMATNEST user's guide.

Appendix S2 BINMATNEST application.

BIOSKETCHES

Miguel A. Rodríguez-Gironés has used evolutionary game theory to study parent–offspring conflict and the evolution of signalling systems. He is currently working on sexual cannibalism and plant–pollinator networks.

Luis Santamaría is interested in evolutionary ecology and plant–animal interactions (herbivory, seed dispersal and pollination), as well as in the interaction between research, policy and management.

Editor: Jose Alexandre Diniz-Filho