Oecologia (2001) 129:281–291
DOI 10.1007/s004420100717

Nicholas J. Gotelli · Gary L. Entsminger

Swap and fill algorithms in null model analysis: rethinking the knight's tour

Received: 30 May 2000 / Accepted: 1 February 2001 / Published online: 31 May 2001

Abstract Community assembly rules are often inferred from patterns in presence-absence matrices. A challenging problem in the analysis of presence-absence matrices has been to devise a null model algorithm to produce random matrices with fixed row and column sums. Previous studies by Roberts and Stone [(1990) Oecologia 83:560–567] and Manly [(1995) Ecology 76:1109–1115] used a "Sequential Swap" algorithm in which submatrices are repeatedly swapped to produce null matrices. Sanderson et al. [(1998) Oecologia 116:275–283] introduced a "Knight's Tour" algorithm that fills an empty matrix one cell at a time. In an analysis of the presence-absence matrix for birds of the Vanuatu islands, Sanderson et al. obtained different results from Roberts and Stone and concluded that "results from previous studies are generally flawed". However, Sanderson et al. did not investigate the statistical properties of their algorithm. Using simple probability calculations, we demonstrate that their Knight's Tour is biased and does not sample all unique matrices with equal frequency. The bias in the Knight's Tour arises because the algorithm samples exhaustively at each step before retreating in sequence. We introduce an unbiased Random Knight's Tour that tests only a small number of cells and retreats by removing a filled cell from anywhere in the matrix. This algorithm appears to sample unique matrices with equal frequency. The Random Knight's Tour and Sequential Swap algorithms generate very similar results for the large Vanuatu matrix, and for other presence-absence matrices we tested. As a further test of the Sequential Swap, we constructed a set of 100 random matrices derived from the Vanuatu matrix, analyzed them with the Sequential Swap, and found no evidence that the algorithm is prone to Type I errors (rejecting the null hypothesis too frequently). These results support the original conclusions of Roberts and Stone and are consistent with Gotelli's [(2000) Ecology 81:2606–2621] Type I and Type II error tests for the Sequential Swap. In summary, Sanderson et al.'s Knight's Tourgenerates large variances and does not sample matrices equiprobably. In contrast, the Sequential Swap generates results that are very similar to those of an unbiased Random Knight's Tour, and is not overly prone to Type I or Type II errors. We suggest that the statistical properties of proposed null model algorithms be examined carefully, and that their performance judged by comparisons with artificial data sets of known structure. In this way, Type I and Type II error frequencies can be quantified, and different algorithms and indices can be compared meaningfully.

Keywords Null model · Assembly rules · Species co-occurrence · Knight's Tour · Presence-absence matrix

Introduction

Assembly rules form one of the most active and controversial research fronts in community ecology (Weiher and Keddy 1999). Although community assembly rules encompass a diverse set of mechanisms, species interactions are still a central focus (Gotelli 1999). Diamond's (1975) assembly rules model suggested that competitive interactions and niche adjustments determine species co-occurrence patterns in replicated island communities. Connor and Simberloff (1979) argued that co-occurrence patterns were no different than expected by chance; their paper popularized the idea of using a null model to test for non-random species co-occurrence patterns.

The data for such a null model consist of a matrix in which each species is a row, and each site is a column. Entries in the matrix indicate the presence (1) or absence (0) of a species in a particular site. In a null model analysis, a presence-absence matrix is reshuffled to produce patterns that would be expected in absence of competi-

N.J. Gotelli (✉)
Department of Biology, University of Vermont,
Burlington, VT 05405, USA
e-mail: ngotelli@zoo.uvm.edu
Tel.: +1-802-6560450, Fax: +1-802-6562914

G.L. Entsminger
Acquired Intelligence Inc., 99 Schillhammer Road,
Jericho, VT 05465, USA

tive interactions (Gotelli and Graves 1996). Connorand Simberloff (1979) suggested that an appropriate set of null matrices are those that preserve the row and column totals of the original matrix. These null matrices thus maintain the same number of species per site (column totals) and the same number of site occurrences per species (row totals) as observed in the original data.

It is a non-trivial problem to devise a computer algorithm to produce an unbiased sample of matrices that maintain row and column totals. Sanderson et al. (1998; Brualdi and Sanderson 1999) recently introduced a novel "Knights Tour" algorithm. The "Knight's Tour" refers to a classic puzzle in computer science: to move a knight across a chessboard so that it visits all 64 squares sequentially, but never lands in the same square twice (Roberts 1986). Sanderson et al. (1998) use the Knight's Tour as a loose metaphor for a "fill" algorithm, in which an empty matrix is filled one cell at a time. At each step, their algorithm seeks exhaustively an open cell that can be filled and will not violate row and column constraints. If no cell can be found that satisfies the constraints, the algorithm backtracks, removing filled cells in order one by one, until it reaches a state from which it can resume filling cells.

Previous workers had used a "Sequential Swap" algorithm, in which the original matrix is reshuffled by repeatedly swapping submatrices that preserve the row and column totals (Roberts and Stone 1990; Manly 1995). Sanderson et al. (1998) compared their results to those of Connorand Simberloff (1979), Roberts and Stone (1990), and Manly (1995) and wrote "We show that results from previous studies are generally flawed". However, Sanderson et al.'s (1998) conclusions are based on the analysis of a single presence-absence matrix (the Vanuatu avifauna). Sanderson et al. (1998) assumed, but did not demonstrate, that the matrices generated by their algorithm represent a random sample of the set of all matrices with fixed row and column totals. In this paper, we examine this assumption and other statistical properties of swap and fill algorithms, and introduce an unbiased version of the Knight's Tour algorithm.

The Results section is divided into three parts. First, we summarize previous analyses of the Vanuatu avifauna presence-absence matrix and compare results using different swap and fill algorithms. Second, we analyze a small data set for which all unique matrices can be calculated by hand, and derive the expected sampling frequencies using Sanderson et al.'s (1998) Knight's Tour and the Sequential Swap algorithms. Finally, we present a newanalysis of a set of random matrices to evaluate the risk of Type I error with the Manly (1995) algorithm. In the Discussion section, we consider the merits of different co-occurrence indices and methods for evaluating thestatistical significance of a null model test. We illustrate these methods with null model analyses of presence-absence matrices for the Vanuatu avifauna, West Indies finches, and Virginia ants. We conclude by summarizing the points of agreement and disagreement with Sanderson et al. (1998).

Materials and methods

Matrix fills

There are two general ways that a matrix can be created that has a set of fixed row and column totals: "swaps" and "fills". In a "swap" algorithm, randomly chosen submatrices of the form:

$$\begin{matrix}0&1\\1&0\end{matrix} \quad \text{or} \quad \begin{matrix}1&0\\0&1\end{matrix}$$

are selected, and the cells in the matrix are swapped. The submatrices are not necessarily formed from adjacent rows and columns; any submatrix of this form can be swapped. Swapping creates a new matrix configuration, but does not alter row and column totals. Swap algorithms always begin with the original matrix and create new matrices by repeatedly swapping.

"Fill" algorithms begin with an empty matrix of 0s and incrementally add randomly chosen 1s. If a state is reached that prevents further fills (because row or column totals would be violated), the algorithm backs up to a previous state and then moves forward again. The algorithm finishes when all 1s have been placed in the matrix and the row and column totals are satisfied. We compared the results of two swap algorithmsand two fill algorithms.

Swap algorithms

Sequential swap

This algorithm used 30,000 initial transpositions of the original matrix to ensure that every submatrix had a good chance of being swapped at least once before consecutive matrices were retained. Then, each subsequent transposition was retained as a different null matrix. Transpositions were created by randomly selecting a pair of rows and a pair of columns in the matrix, then swapping the four cells if possible. If a swap was not possible, another pair of random rows and columns was selected. This procedure is similar to that used by Manly (1995).

Independent swap

This algorithm did not retain consecutive matrices. Instead, each null matrix was created through 30,000 independent swaps of the original matrix. The procedure was repeated for each null matrix, beginning with the observed matrix as the starting point, similar to the method introduced by Roberts and Stone (1990). This procedure should theoretically produce the entire set of matrices possible for a given set of row and column totals (Brualdi 1989), and it provides a good check for the independence of the sequential matrices thatare retained in the Sequential Swap.

Fill algorithms

Random Knight's Tour

In Sanderson et al.'s (1998) Knight's Tour, an empty matrix is filled by sequentially adding one randomly chosen cell at a time. At each step in the Knight's Tour, the algorithm exhaustively samples each node, searching for a cell that can be filled that will still satisfy all of the row and column constraints. If no cell can be found, the algorithm backtracks one step, by removing the most recently filled cell, and then beginning the search again. The procedure is repeated until the entire matrix is filled.

We found it difficult to program this algorithm efficiently for largematrices because of the time needed to update lists and keep track of all of the nodes that were sampled and visited. Instead, we used a simpler approach. We reasoned that if the Knight's Tour was a robust solution to the matrix fill problem, an algorithm that samples nodes randomly and retreats randomly should pro-

Table 1 Null model analyses of the S^2 metric for the presence-absence matrix of Vanuatu avifauna (Diamond and Marshall 1976). In all studies, the observed S^2 metric=148.85. Standardized effect size (SES)=(Observed S^2 – $\bar{x}$ Simulated)/ σ^2 Simulated). *Low* and *High* indicate extreme values generated by the simulation. *n.r.* Not reported by original authors. See text for description of simulation algorithms

Source	Algorithm	Replicates	$\bar{x}$ Simulated	σ^2 Simulated	SES	Low	High	Upper-tail *P*
Sanderson et al. (1998)	Knight's Tour	5,000	150.47	1.242	–1.76	n.r.	203.65	n.r
Roberts and Stone (1990)	Sequential Swap	1,000	147.10	0.0529	7.60	n.r.	147.79	<0.001
Current study	Sequential Swap	1,000	147.84	0.16234	2.49	146.97	149.41	0.039
Current study	Sequential Swap	5,000	147.84	0.13802	2.61	146.82	149.12	0.0042
Current study	Independent Swap	1,000	147.92	0.15738	2.32	146.86	149.04	0.012
Current study	Random Knight's Tour	1,000	147.90	0.13353	2.58	146.85	149.15	0.006
Current study	Exhaustive Knight's Tour	1,000	149.81	0.32620	–1.68	148.20	151.50	0.949

duce similar results to Sanderson et al.'s (1998) algorithm, which samples nodes exhaustively and retreats sequentially. As with Sanderson et al.'s (1998) Knight's Tour, our Random Knight's Tour begins with an empty matrix and fills it by sequentially adding randomly chosen cells and backtracking when necessary until a solution is reached. In our Random Knight's Tour, the algorithm does not sample exhaustively at each node, but only examines a small number of randomly chosen cells for filling. If a cell cannot be filled, our algorithm retreats by randomly removing a filled cell from anywhere in the matrix, not necessarily the last cell filled. Although not mentioned by Sanderson et al. (1998), Stone and Roberts (1990) also reported on the results of a Random Knight's Touralgorithm (the "Milne method") that gave similar results to their swap algorithm (L. Stone, personal communication).

For small matrices, we sampled a single randomly chosen cell at each node. If a usable cell could not be found, the algorithm immediately retreated one step. For large matrices, this algorithm was inefficient because the program spent too much time backtracking. For the Vanuatu matrix, we analyzed a range of sampling intensities (Table 2). The lowest sampling intensity of 444 cells (~ 50% of the open cells) allowed us to create 1,000 random matrices in approximately 12 h of simulation time.

Exhaustive Knight's Tour

For this algorithm, we increased the sampling intensity to avery large number, so that the available cells at each node were exhaustively searched. For the Vanuatu matrix, we sampled 8,888 cells at each node, ensuring that all available cells were checked before the algorithm backtracked. Our Exhaustive Knight's Tour is identical to Sanderson et al.'s (1998) algorithm, except that we did not retreat sequentially when backtracking. As we show, this algorithm generates results that are qualitatively similar to those of Sanderson et al. (1998), although we did not observe the very large variances generated by their algorithm.

Our algorithms were programmed in Delphi Version 4.0, and implemented in EcoSim Version 5.0 (Gotelli and Entsminger 2000). The on-line version of EcoSim contains the Sequential Swap, Independent Swap, and Random Knight's Tour algorithms, and allows for the choice of the *C*-score, the number of checkerboards, the number of species combinations, and the variance ratio (Schluter 1984) as co-occurrence indices. The on-line version also contains data matrices for the West Indian finches and Virginia ants that are describedin this paper. A special compiled version of EcoSim that analyzes the S^2 index and contains the Exhaustive Knight's Tour is available by request from the authors, as is the Vanuatu data matrix.

Data matrices

Using the original data in the appendix of Diamond and Marshall (1976), we reconstructed the data matrix for birds of the Vanuatu (= New Hebrides) Archipelago (56 species × 28 islands). Our matrix row and column totals matched exactly those reported in Table 1 of Sanderson et al. (1998), as did the reported S^2 index of 148.85. We analyzed two other real data matrices, the finches of the West Indies (17 species × 19 islands; Gotelli and Abele 1982), and a small-scale pitfall trap survey of ants from a Virginia woodland (11 species× 12 sites; Gotelli and Arnett 2000).

Co-occurrence indices

Roberts and Stone (1990) introduced the S^2 index to measure the number of shared sites between species pairs. For each unique pair of species, the number of shared sites *S* is tabulated, squared, and summed. A closely related index that we prefer is the *C*-score (Stone and Roberts 1990). The *C*-score measures the average number of "checkerboard units" of the form:

$$\begin{array}{ccc} 01 & & 10 \\ & \text{or} & \\ 10 & & 01 \end{array}$$

for each unique species pair. Note that these checkerboard units form the submatrices that are exchanged in the swap algorithms. Stone and Roberts (1990) showed that the number of checkerboard units for (CU) any species pair can be calculated as:

$$CU=(r_i-S)(r_j-S) \tag{1}$$

where r_i and r_j are the row totals, and *S* is the number of sites occupied by both species. The *C*-score is the average CU calculated for all unique pairs of species.

Results

Vanuatu analyses

Sanderson et al. (1998) tested the S^2 index on the Vanuatu matrix and compared their result to the test of Roberts and Stone (1990). For 1,000 simulations, Roberts and Stone (1990) reported a mean index of 147.10, with a sample variance (labeled as a standard deviation by Sanderson et al. (1998)) of 0.0529. The observed S^2 metric (148.85) was significantly larger than that generated by the sequential swap algorithm, and Roberts and Stone (1990) concluded that the pattern was non-random. Using their Knight's Touralgorithm, Sanderson et al. (1998) simulated 5,000 matrices and found an average S^2 metric of 150.47, with a sample variance of 1.242. The observed index did not differ significantly from that generated by the Knight's Tour algorithm. It is on the basis of this result that Sanderson et al. (1998) concluded that the S^2 metric is flawed and the randomization method of Manly (1995) is biased.

However, Sanderson et al.'s (1998) results are not directly comparable to Roberts and Stone's (1990), in part because Sanderson et al. (1998) used 5,000 simulations, whereas Roberts and Stone (1990) used 1,000. More important, Robertsand Stone (1990) imposed an additional constraint in their simulations that does not appear to have been incorporated by Sanderson et al. (1998). In their analysis, Roberts and Stone (1990) maintained the so-called "incidence function" of each species, so that the simulated range of island sizes occupied by each species matched that in the observed data set. This additional constraint means that the matrices created by Roberts and Stone (1990) constituted a subset of all random matrices that maintained row and column totals.

These different procedures do account for some, but not all, of the differences between the results of Sanderson et al. (1998) and Roberts and Stone (1990). When we analyzed the Vanuatu matrix with the Sequential Swap algorithm, we also obtained a significant result (P=0.039), though not as extreme as that reported by Roberts and Stone (1990; P<0.001). Our results from the Sequential Swap matched the results of the Independent Swap, in which each matrix is constructed from a series of Independent Swap s (Table 1).

Our Random Knight's Tour generated results that were very similar to those of the Sequential and Independent Swap. For a small data set, we confirmed that the Random Knight's Tourdoes generate a non-biased sample of random matrices (Appendix). In contrast, our Exhaustive Knight's Tour generated results very different from the Random Knight's Tour and the two Swap algorithms (Table 1). As reported in Sanderson et al.'s (1998) study, our Exhaustive Knight's Tour led to a shift in the mean and an increase in the variance, though not as extreme as Sanderson et al. (1998) found with their algorithm.

The behavior of the Knight's Tour fill algorithms are very sensitive to the amount of sampling at each node. If exhaustive sampling is used, the results tend towards those reported by Sanderson et al. (1998). With less exhaustive sampling, the results converge towards those of the swap algorithms (Table 2).

Type I error analysis

Repeated swapping of submatrices will theoretically produce the full set of null matrices (Brualdi 1980), and we note that the Swap algorithms and the Random Knight's Tour produce consistent results. In their criticism of the Sequential Swap algorithms, Sanderson et al. (1998) caution that "it cannot be guaranteed that cyclic perturbations can be avoided. Thus, it is possible to perform two or more perturbations and end up with the same matrix".

However, cyclic perturbations are unlikely for most real matrices. For example, in the Vanuatu matrix, there are initially 14,676 different submatrices that can be swapped at the first step of the Sequential Swap procedure. Thus, the chance of reversing a forward step and generating the same matrix configuration is 1/14,676= 6.8×10^{-5}. We agree with Sanderson et al. (1998) that there are many possible matrix rearrangements that can be created for the Vanuatu matrix (and most real presence-absence matrices) that satisfy row and column totals. Therefore it is unlikely that cyclic perturbations are a serious problem for the Sequential Swap algorithm.

Sanderson et al. (1998) rejected the Sequential Swap algorithm because it gave different results for the Vanuatu presence-absence matrix than did their Knight's Tour algorithm. However, it is not logical to evaluate different algorithms by comparing their behavior with a real presence-absence matrix – such a matrix will contain unknown amounts of biological structure and randomness. Instead, we should compare the performance of algorithms on artificial data sets with known properties (e.g., Gotelli et al. 1997; Garvey et al. 1998; Shenk et al. 1998).

By claiming that Roberts and Stone's (1990) finding of a non-random S^2 metric was incorrect, Sanderson et al. (1998) invoked a Type Ierror, in which a true null hypothesis is incorrectly rejected. To attack this problem directly, we use artificial data sets to decide whether Roberts and Stone (1990) incorrectly rejected the null hypothesis for the Vanuatu matrix. We first created 100 random matrices in which we were sure there should be no pattern present. We wanted these matrices toresemble the Vanuatu matrix, but with no trace of biological structuring.

To create such matrices, we began with the original Vanuatu matrix and then reshuffled the cells within each row. Reshuffling the cells within each row eliminates any differencesamong sites due to area, resources, or

Table 2 Effects of sampling intensity on performance of the Knight's Touralgorithm. For the Vanuatu matrix, results are given for 1000 simulations. Sampling intensity indicates the number of cells the algorithm samples before backtracking. Other variables as in Table 1

Sampling intensity	$\bar{x}$ Simulated	σ^2 Simulated	SES	Low	High	Upper-tail P
444 (Random Knight's Tour)	147.90	0.13353	2.58	146.85	149.15	0.006
888	147.95	0.16846	2.18	146.79	149.83	0.029
1,600	148.85	0.42103	0.51	146.94	150.77	0.276
8,888 (Exhaustive Knight's Tour)	149.81	0.32620	–1.68	148.20	151.50	0.949

habitat availability. This is analogous to a colonization model in which all sites are equiprobable, and each species selects a site independently of the presence of any other species. We stress that the creation oftest matrices in this way is not circular, because it is a completely different filling algorithm than either the swap or fill algorithms.

The resulting set of 100 matrices should not exhibit any non-randomness as a group, although we would expect that, by chance, approximately 5% of these matrices would generate significant results in a well-behaved statistical analysis. Running more than 100 matrices in this way would have been desirable, but the test is time-consuming, and the error frequency shouldnot be appreciably different for 100 vs 1,000 matrices.

Next, we analyzed each random matrix with the Sequential Swap algorithm, completing 5,000 iterations and performing separate analyses for the S^2 and the *C*-score indices. Thus, although the column totals for each random matrix could vary freely as the matrix was being constructed, the resulting column and row totals were fixed in the Sequential Swap algorithm. For each matrix, we recorded the upper-tail probability that the observed index was more extreme than predicted by the null model. If Sanderson et al.'s (1998) claim of a Type I error is correct, we should find that many more than 5 of the 100 random matrices caused the null hypothesis to be rejected. On the other hand, if 5 or fewer of the 100 random matrices caused the null hypothesis to be rejected at the 0.05 significance level, there would be no indication of an excessive Type I error rate, and Roberts and Stone's (1990) original conclusion would be supported.

Table 3 gives the ordered upper-tail probabilities for the 15 most extreme random matrices. For the S^2 metric, 5 out of 100 random matrices rejected the null hypothesis at P=0.05, and 12 out of 100 matrices rejected the null hypothesis at P=0.10. The calculated P value for the actual Vanuatu matrix was 0.0042. This value was more extreme than 98 of the 100 random matrices, suggesting that the probability of a Type I error for the S^2 index of the Vanuatu matrix was less than 0.02.

Table 3 Ordered tail probabilities for the 15 most extreme matrices from a set of 100 random matrices. Each matrix was tested using the Sequential Swap algorithm with5,000 replicates. The probability value for the real Vanuatu matrix is underlined and is also shown in sequence. Results are shown for both the *C*-score and the S^2 indices. For both indices, the null hypothesis was rejected at P=0.05 for only 5 of the 100 random matrices, indicating good Type I error behavior

Matrix no.	Upper-tail P (S^2 index)	Matrix no.	Upper-tail P (*C*-score index)
68	0.0038	Vanuatu	0.0006
25	0.0040	62	0.0066
Vanuatu	0.0042	92	0.0096
62	0.0062	63	0.0268
50	0.0412	65	0.0360
41	0.0486	50	0.0366
90	0.0548	10	0.0578
73	0.0714	73	0.0590
65	0.0758	46	0.0652
46	0.0794	11	0.0776
91	0.0874	14	0.0802
78	0.0940	78	0.0888
20	0.0922	31	0.0922
60	0.1066	70	0.1006
95	0.1090	91	0.1158
88	0.1166	42	0.1172

For the *C*-score, 5 out of the 100 random matrices rejected the null hypothesis at P=0.05, and 12 out of 100 random matrices rejected the null hypothesis at P=0.10. The calculated P value for the actual Vanuatu matrix was P=0.0006. This value was more extreme than the tail probability for all 100 of the random matrices, suggesting that the probability of a Type I error for the *C*-score metric of the Vanuatu matrix was less than 0.01.

Thus, we find no evidence that Roberts and Stone (1990) made a Type I error when they rejected the null hypothesis for the S^2 metric of the Vanuatu matrix. When the Sequential Swap algorithm is used on a set of random matrices, it correctly generates a non-significant upper-tail pattern 95% of the time. In contrast, the S^2 and *C*-score indices for the actual Vanuatu matrix are significantly larger than expected by chance, and the pattern is more extreme than in most random matrices that have identical row totals. These analyses of random matrices support the original conclusions of Roberts and Stone (1990). The S^2 metric for the Vanuatu matrix is significantly larger than chance, and the Sequential Swap method does not have an excessive Type I error rate.

Discussion

Which algorithm should be used?

Our results suggest that the Knight's Tour algorithm of Sanderson et al. (1998) should not be used for null model analysis. This algorithm does not sample matrices equiprobably (Appendix), and the results it produces rest delicately on the assumption of exhaustive sampling at each node (Table 2). The Type I and Type II error properties of Sanderson et al.'s (1998) Knight's Tour are unknown, and the anomalous result it generates for the Vanuatu matrix (Table 1) is not a valid basis for claiming that "results from previous studies are generally flawed".

In contrast, the Sequential Swap has good Type I and Type II error properties (Gotelli 2000), and there is no evidence that it causes the null hypothesis to be rejected improperly for the Vanuatu matrix (Table 3). Although the Sequential Swap shows good statistical performance, it also has the potential to generate matrices with unequal frequencies (Appendix). Zaman and Simberloff (unpublished data) addressed this problem with a Weighted Swap algorithm, in which the weights are inversely proportional to the frequencies with which random matrices are visited by swapping. For the Vanuatu matrix, this Weighted Swap also generated highly nonrandom results (P=10^{-6}), and results of the Sequential and Weighted swaps were similar for another presence-absence matrix they tested. G. Cobb (Department of Mathematics, Mt. Holyoke College) found that small runs of 1,000 replicates of the Sequential Swap were significantly heterogeneous and autocorrelated, due to similarity of sequential matrices generated by the algorithm. However, Cobb also found that a set of random matrices generated by the Sequential Swap from the presence-absence matrix of Darwin's finches had a uniform probability distribution, as it should for an unbiased test. Collectively, these results suggest the Sequential Swap performs well in empirical tests, and its behavior on the Vanuatu matrix converges with that of the Independent Swap, Zaman and Simberloff's (unpublished data) Weighted Swap, and our Random Knight's Tour.

How should indices be tested?

Sanderson et al. (1998) followed the lead of Connor and Simberloff (1979) and plotted the number of species pairs sharing 0,1,2...*n* islands. They tested for deviations of observed and expected values in each category by constructing a parametric 99% confidence interval for a Student's *t*-test. Observations that fell outside of this interval were classified as significant. This procedure is very similar to the original chi-square test that Connor and Simberloff (1979) used to evaluate the same distribution of data. As Roberts and Stone (1990)thoroughly explained, a parametric chi-square test is inappropriate in this case because the set of shared island numbers are not independent observations. Sanderson et al. (1998) revived this error by using a *t*-test and evaluatingall of the species pair combinations. No appeals to parametric theory can get around the fact that these numbers are not independent. Indeed, one of the major reasons for using randomization methods in the first place is to avoid problems of exactly thissort.

To test the significance of patterns in null model analysis, we prefer the methods that are used in classic randomization tests (Edgington 1987; Manly 1991): describe the pattern in the matrix with a single index, and compare the observed value of that index directly to the distribution of index values from the simulated matrices. Extreme values of the observed index in the tails of the distribution indicate statistically significant patterns.

Which co-occurrence index should be used?

The utility of species co-occurrence indices should be based not only on their statistical properties, but on their relationship to ecological theory. Sanderson et al.'s (1998) used the individual deviations of species-pair classes as their index. Thus, an assembly rule derived from their analysis of the Vanuatu matrix (their Fig. 2) would be: "There are fewer avian species pairs occupying 2 islands and 9 islands, and more species pairs occupying 10 islands, than expected by chance in the Vanuatu Archipelago". We do not see how meaningful biological interpretations could be attached to such an assembly rule.

In contrast, we prefer three different indices that address Diamond's (1975) assembly rules model: the number of species combinations (Pielou and Pielou 1968), the number of species pairsforming perfect checkerboard (Diamond 1975), and Stone and Robert's (1990) *C*-score. A fourth metric, Schulter's (1984) variance ratio, is also a useful measure of species covariance. However, the variance ratio is determined solely by the row and column sums of the matrix, so it cannot be tested using the randomization algorithms described in this paper, all of which preserve row and column sums (Gotelli 2000).

Pielou and Pielou (1968) first introduced the number of species combinations as an index of community structure. This index is directly related to Diamond's (1975) first and second assembly rules:

> *"1. If one considers all the combinations that can be formed from a group of related species, only certain ones of these combinations exist in nature."*
>
> *"2. These permissible combinations resist invaders that would transform them into a forbidden combination."*

If assembly rules 1 and 2 are met, a set of islands or sites should harbor significantly fewer species combinations than expected by chance.

A second useful index is the number of species pairs that never co-occur, forming "checkerboard" distributions. This index describes Diamond's (1975) fifth assembly rule:

> *"5. Some pairs of species never coexist, either by themselves or as part of a larger combination."*

If this assembly rule is in operation, there should be significantly more species pairs in a matrix forming perfect checkerboards than expected by chance. Note that this number is equivalent to the number of species sharing 0 islands in Connor and Simberloff (1979) and Sanderson et al. (1998).

A third useful index is Stone and Robert's (1990) *C*-score. This index also measures the degree to which species co-occur, but it is not as stringent as the checkerboard measurement because it does not require perfect segregation between species. For a community structured by species interactions, the *C*-score should be significantly larger than expected by chance. Diamond's (1975) other assembly rules (nos. 3, 4, 6, and 7) are more difficult to test with simple null models because they involve complex comparisons of patterns in species-rich and species-poor communities.

Of course, the demonstration of non-random patterns in the *C*-score, number of species combinations, or the number of checkerboards does not, by itself, constitute a proof of Diamond's (1975) assembly rules model (Connor and Simberloff 1986). Many other factors, including habitat heterogeneity, dispersal barriers, and positive interactions between species can also lead to non-randomness in these indices (Weiher and Keddy 1999). However, the null model analysis is the important first step towards at least establishing whether the patterns are random or not.

Although the *C*-score, number of checkerboards, and number of species combinations are closely related, they measure different patterns in a matrix and have different statistical properties. Gotelli (2000) has carried out simulation tests with these indices andcompared their performance on random matrices and on structured matrices that have differing levels of random "noise" added to the co-occurrence patterns.

The *C*-score, when used in combination with the Sequential Swap algorithm, has good statistical power for detecting pattern in structured matrices that have a considerable amount of random noise added to them. At the same time, it has good Type I error properties and does not lead to an excessive rejection of the null hypothesis when tested with random matrices (Gotelli 2000). A low frequency of Type I errors for the *C*-score and the Sequential Swap algorithm are confirmed by our additional analyses of the Vanuatu matrix in this paper (Table 3).

Empirical examples

To illustrate utility of the methods we propose, we analyzed three presence-absence matrices with the Sequential Swap algorithm and present patterns based on the *C*-score, the number of checkerboard pairs, and the number of species combinations. We used 5,000 iterations in each analysis. Results were similar when these matrices were analyzed with the Random Knight's Tour.

For the Vanuatu avifauna, the *C*-score was significantly greater than expected by chance (Fig. 1a), but the number of checkerboard pairs (Fig. 1b) and the number of species combinations (Fig. 1c) did not differ from random. For the number of species combinations, we note that each of the 28 islands in the Vanuatu matrix supported a different species combination. Similarly, in all 5,000 simulated communities, species combinations were never repeated for any pair of islands, so 28 combinations were observed in every case.

For the West Indies fiches, the *C*-score was also larger than expected by chance (Fig. 2a). Only 10 unique species combinations occurred on the 19 islands, which was significantly less than expected by chance (Fig. 2b), confirming Diamond's (1975) 1st and 2nd assembly rule patterns. The number of checkerboard pairs did not differ from random (Fig. 2c).

For the Virginia ants, none of the three co-occurrence indices were different from random (Fig. 3a–c). Because the ant matrix was created from a grid of pitfall traps that sampled species at a very small spatial scale (5×5 m), we would not have expected to see much structure in comparison to the avifauna of island archipelagos.

In conclusion, we are in agreement with Sanderson et al. (1998) on several major issues. First, we agree that the entire issue of testing for non-randomness in species co-occurrence patterns is important, especially with the continued interest in community assembly rules (Weiher and Keddy 1999) and computer simulation models (Hilborn and Mangel 1997; Cipra 2000). Second, we agree that an algorithm that maintains observed row and column sums is useful for testing community patterns, although this is by no means the only valid null model (Gotelli 2000). Wealso agree that, for most real matrices, there are many different matrix rearrangements that can be constructed that satisfy the row and column constraints.

However, there are also some major areas of disagreement. Our results suggest that Sanderson et al.'s (1998) Knight's Touris a biased algorithm that should not be used for null model analysis. We think it is inappropriate to use a *t*-test on non-independent data, and that deviations from the shared-species distribution cannot be interpreted biologically. We prefer a simple randomization test for assessing statistical significance. We recommend the *C*-score, number of species combinations, and number of

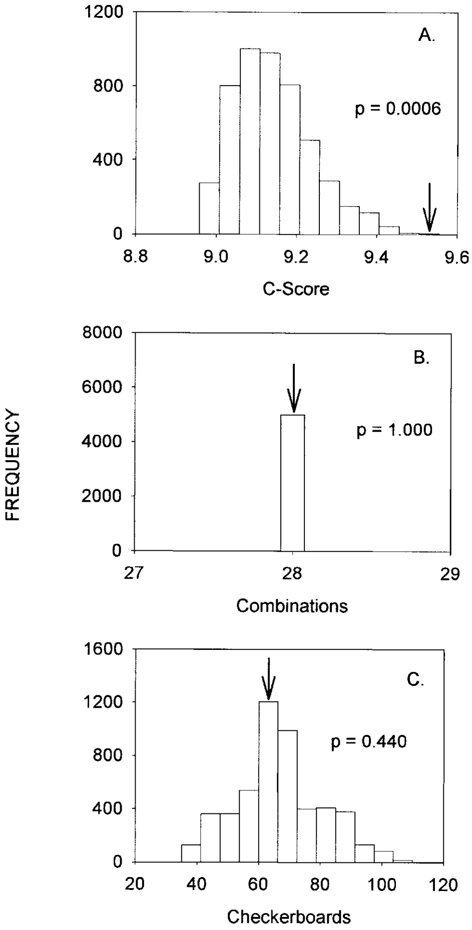

Fig. 1a–c Null model analyses of the Vanuatu avifauna. Each histogram shows the range of values from 5,000 simulated assemblages. The arrow indicated the observed value, and p is the one-tailed probability. **a** *C*-score; **b** number of species combinations; **c** number of species pairs forming perfect checkerboards

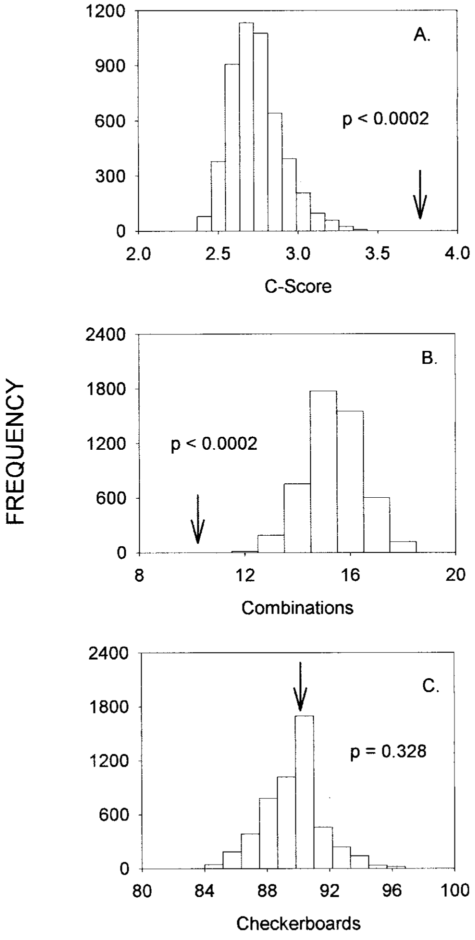

Fig. 2 Null model analyses of West Indies finches. Labels as in Fig. 1

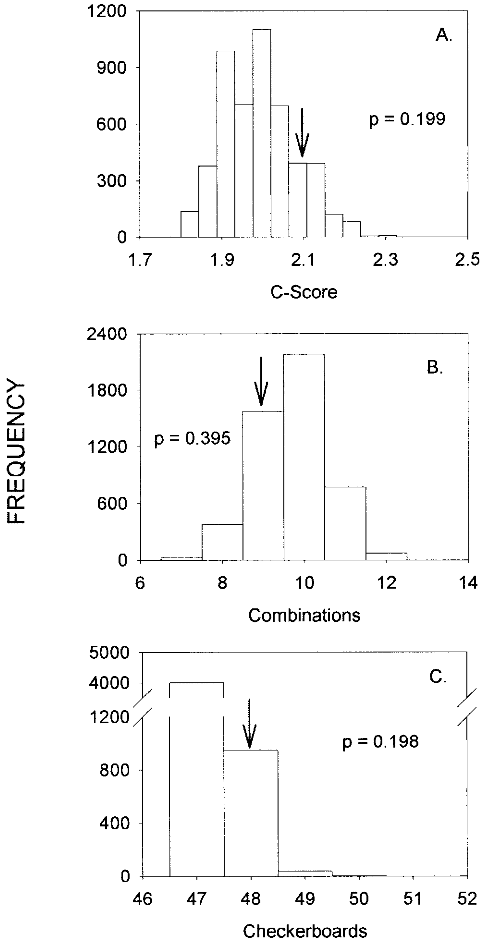

Fig. 3 Null model analyses of small-scale Virginia ant sample. Labels as in Fig. 1

checkerboard pairs as three indices with good statistical properties that are directly relevant to Diamond's (1975) assembly rules model. Finally, an error analysis of random matrices supports Roberts and Stone's (1990) original conclusions: the S^2 metric for the Vanuatu matrix is larger than expected by chance. There is no evidence that this pattern is caused by a Type I error because random matrices tested with the Sequential Swap algorithm rejected the null hypothesis 5% of the time (Table 3).

This paper represents the latest contribution to a controversy over the statistical analysis of the Vanuatu presence-absence matrix that has lasted for over 20 years (e.g., Connor and Simberloff 1979,1983; Diamond and Gilpin 1982; Wilson 1987; Roberts and Stone 1990; Sanderson et al. 1998; Zaman and Simberloff, unpublished data).No wonder some ecologists are uneasy about the use of null models! Future exchanges can be minimized by carefully examining the statistical properties of null models, and systematically testing their behavior with artificial data sets of known structure. Such analyses are tedious and not very exciting, but they are necessary if null models are to serve as reliable statistical tools for ecologists and biogeographers.

Acknowledgements We thank A. Ellison, G. Cobb, P.Peres-Neto, J. Sanderson, D. Simberloff, L. Stone, and the University of Vermont Ecology Discussion Group for comments on the manuscript. We thank Inprise Corporation/Borland for Delphi development software. EcoSim software development is supported by NSF grants BIR-9612109 and DBI 9725930.

Appendix

Expected sampling frequencies of swap and fill algorithms

Zaman and Simberloff (unpublished data) note that Sanderson et al. (1998) have not demonstrated that the Knight's Toursamples all matrices uniformly. Here we analyze a simple example in which it is possible to calculate by hand the probability of reaching different matrix solutions using several swap and fill algorithms. This analysis demonstrates that the Knight's Tourdoes not select matrices equiprobably.

We begin with a small 3×3 matrix. We label the cells of this matrix 1–9 consecutively:

$$\begin{matrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{matrix}$$

Suppose that we wish to construct random matrices with four cells filled. The matrix constraints we use are row totals of (1, 2, 1) and column totals of (1, 2, 1). There are

$$\binom{9}{4}=126$$

126 unique ways to fill a 3×3 matrix with four 1s. Of these 126 matrices, exactly 5 matrices satisfy the specified row and column constraints. We label these matrices A-E:

Matrix A

$$\begin{matrix}0 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 0\end{matrix}$$

Matrix B

$$\begin{matrix}0 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 1\end{matrix}$$

Matrix C

$$\begin{matrix}1 & 0 & 0\\ 0 & 1 & 1\\ 0 & 1 & 0\end{matrix}$$

Matrix D

$$\begin{matrix}0 & 0 & 1\\ 1 & 1 & 0\\ 0 & 1 & 0\end{matrix}$$

Matrix E

$$\begin{matrix}0 & 1 & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\end{matrix}$$

An unbiased null model algorithm should generate all five matrices with frequency f=0.20.

Exhaustive Knight's Tour

The probabilities for the Exhaustive Knight's Tour can be calculated by hand for this small example. We illustrate the method with matrix A, which is the easiest to compute. Matrix A has entries in cells 2,4,6, and 8. There are 4!=(4)(3)(2)(1)=24 sequences by which this matrix can be reached filling sequentially with the exhaustive knight's tour.

We calculate the probabilities of reaching each particular sequence through the Exhaustive Knight's Tour, and then sum those probabilities to obtain the total frequency with which the matrix is reached. Suppose we wish to calculate the probability that Matrix A is reached by filling in sequence the cells 2, 4, 6, and 8. Thus, we seek to calculate *P*(Matrix A |2468).

For the initial placement of cell 2 in an empty matrix, the probability is (1/9).

Once cell 2 has been selected, matrices C and D are eliminated from the solution space. For the next cell selection, cell 2 cannot be chosen again because it is already filled. Moreover, cells 1 and 3 are also not possible because these would violate row or column constraints. Thus, on the next step of the exhaustive knight's tour, there are 6 possible cells that can be filled (cells 4, 5, 6, 7, 8, or 9), and the probability of choosing cell 4 is (1/6).

Now that cells 2 and 4 have been chosen, only matrices A and B remain in the solution space. The only cells now available for filling that havenot already been selected (2, 4) or that do not violate the row and column totals (1, 3, 7) are cells 5, 6, 8, and 9. Thus, on the third step of the Exhaustive Knight's Tour, the probability of choosing cell 6 is (1/4).

Now that cells 2, 4, and 6 have been chosen, cell 8 is the only one available that will satisfy the matrix constraints, and this completes the construction of Matrix A.

Therefore, the probability that Matrix A is created by filling cells 2, 4, 6 and 8 in sequence is: *P*(Matrix A |2468)=(1/9)(1/6)(1/4)(1)=(1/216) .

Notice that for the fill sequence (2, 4, 6, 8), the first two cell fills (2, 4) are diagonal to one another. A different pathway occurs if the first two cell fills are opposite of one another. For example, consider the fill sequence (2, 8, 4, 6). As before, the probability of selecting in sequence cell 2 and then cell 8 is (1/9)(1/6). However, with cells 2 and 8 filled, there are now only two possibilities at the third step (4 or 6). Therefore, the probability that Matrix A is created by filling cells 2, 8, 4, and 6 in sequence is: *P*(Matrix A |2846)=(1/9)(1/6)(1/2)(1)=(1/108)

Of the 24 sequences that lead to Matrix A, 16 of them have the "diagonal" pattern for the first two cells in the sequence, and 8 of them have the "opposite" pattern. Therefore, the overall probability of reaching Matrix A by the Exhaustive Knight's Tour Knight's Tour is: *P*(Matrix A)=(16)(1/216)+(8)(1/108)=0.14815.

If the Exhaustive Knight's Tour Knight's Tour were unbiased, then *P*(Matrix A)=0.2000 (Table 4).

Matrix A is reached less frequently than other matrices using the Exhaustive Knight's Tour Knight's Tourbecause if the first cell chosen randomly is in the center (5) or any of the corners (1, 3, 7, or 9), Matrix A is immediately eliminated from the solution space.

The calculations for Matrices B-E are more complicated, because there are several different probability pathways represented among the set of 24 sequences. However, using the same methods, it can be shown that

Table 4 Expected and observed frequencies unique matrices generated by the Random Knight's Tour, the Exhaustive Knight's Tour, and the Sequential Swap. An unbiased algorithm should produce all five matrices with equal frequency

Matrix	Ideal frequency	Random Knight's Tour	Sequential Swap frequency	Sequential Swap frequency	Knight's Tour frequency	Exhaustive Knight's Tour
	(Expected)	(n=1,000)	(Expected)	(n=1,000)	(Expected)	(n=1,000)
A	0.2000	193	0.2500	237	0.14815	151
B	0.2000	201	0.1875	193	0.21296	222
C	0.2000	207	0.1875	181	0.21296	207
D	0.2000	202	0.1875	198	0.21296	211
E	0.2000	197	0.1875	191	0.21296	209

P(Matrix B)=0.21296. Matrices B–E are oversampled by the Exhaustive Knight's Tour Knight's Tourbecause there are more starting positions from which these matrices can be reached. Note that the patterns in Matrices B–E can all be produced by reflection or transposition, and the probabilities of reaching these matrices turn out to be identical. We also confirm that 0.14815+(4)(0.21296)=1.0000, as it should.

Sequential Swap

To calculate the expected frequencies for the Sequential Swap, we used a different approach. For each matrix A–E, a single swap will produce one of the other matrices. Matrix A can generate any of the other four matrices through a single swap, whereas matrices B–E can produce only three possible matrices through swapping. These probabilities can be expressed in a transition matrix:

	Initial matrix					
		A	B	C	D	E
Swapped matrix	A	0	0.33	0.3	0.33	0.33
	B	0.25	0	0	0.33	0.33
	C	0.25	0	0	0.33	0.33
	D	0.25	0.33	0.33	0	0
	E	0.25	0.33	0.33	0	0

Because a swap by definition produces a different matrix configuration, the diagonals of this matrix must equal zero. The eigenvector solution gives the expected frequencies of the different matrices produced by swapping many times (Table 4). In contrast to the Exhaustive Knight's Tour, the Sequential Swap is expected to oversample Matrix A (P=0.2500) and undersample Matrices B–E (P=0.1875). This pattern arises because there are more ways to reach Matrix A by swapping than to reach Matrices B–E.

Simulated matrix frequencies

To confirm our probability calculations, we used the Sequential Swap, Exhaustive Knight's Tour, and Random Knight's Tour algorithms to create 1,000 3×3 matrices, and kept track of the frequency with which matrices A-E were produced. Both the Sequential Swap and the Exhaustive Knight's Tour observed frequencies were in good agreement with the theoretically derived expectations (sequential swap $X^2_4=1.72$, $P=0.78$; Exhaustive Knight's Tour $X^2_4=0.63$, $P=0.96$). The Exhaustive Knight's Tour produced frequencies that deviated significantly from an unbiased uniform distribution ($X^2_4=15.68$, $P=0.003$). The Sequential Swap produced frequencies that deviated weakly ($X^2_4=9.32$, $P=0.054$), whereas the Random Knight's Tour showed a very good fit to the unbiased uniform distribution ($X^2_4=0.56$, $P=0.97$).

References

Brualdi RA (1989) Matrices of zeroes and ones with fixed row and column sum vectors. Lin Algebr Appl 33:159–231

Brualdi RA, Sanderson JG (1999) Nested species subsets, gaps, and discrepancy. Oecologia 119:256–264

Cipra B (2000) Revealing uncertainties in computer models. Science 287:960–961

Connor EF, Simberloff D (1979) The assembly of species communities: chance or competition? Ecology 60:1132–1140

Connor EF, Simberloff D (1983) Interspecific competition and species co-occurrence patterns on islands: null models and the evaluation of evidence. Oikos 41:456–465

Connor EF, Simberloff D (1986) Competition, scientific method, and null models in ecology. Am Sci 74:155–162

Diamond JM (1975) Assembly of species communities. In: Cody ML, Diamond JM (eds) Ecology and evolution of communities. Harvard University Press, Cambridge, pp 342–444

Diamond JM, Gilpin ME (1982) Examination of the "null" model of Connor and Simberloff for species co-occurrence on islands. Oecologia 52:64–74

Diamond JM, Marshall AG (1976) Origin of the New Hebridean avifauna. Emu 76:187–200

Edgington ES (1987) Randomization tests. Dekker, New York

Garvey JE, Marschall EA, Wright RA (1998) From star charts to stoneflies: detecting relationships in continuous bivariate data. Ecology 79:442–447

Gotelli NJ (1999) How do communities come together? Science 286:1684–1685

Gotelli NJ (2000) Null model analysis of species co-occurrence patterns. Ecology 81:2606–2621

Gotelli NJ, Abele LG (1982) Statistical distributions of West Indian land bird families. J Biogeogr 9:421–435

Gotelli NJ, Arnett AE (2000) Biogeographical effects of red fire ant invasion. Ecol Lett 3:257–261

Gotelli NJ, Entsminger GL (2000) EcoSim: Null models software for ecology Version 5.0. Acquired Intelligence Inc. & Kesey-Bear, Burlington, Vt. http://homepages.together.net/~gentsmin/ecosim.htm

Gotelli NJ, Graves GR (1996) Null models in ecology. Smithsonian Institution, Washington, D.C.

Gotelli NJ, Buckley NJ, Wiens JA (1997) Co-occurrence of Australian land birds: Diamond's assembly rules revisited. Oikos 80:311–324

Hilborn R, Mangel M (1997) The ecological detective: confronting models with data. Princeton University Press, Princeton

Manly BFJ (1991) Randomization and Monte Carlo methods in biology. Chapman and Hall, London

Manly BFJ (1995) A note on the analysis of species co-occurrences. Ecology 76:1109–1115

Pielou DP, Pielou EC (1968) Association among species of infrequent occurrence: the insect and spider fauna of *Polyporus betulinus* (Bulliard) Fries. J Theor Biol 21:202–216

Roberts A, Stone L (1990) Island-sharing by archipelago species. Oecologia 83:560–567

Roberts E (1986) Thinking recursively. Wiley, New York

Sanderson JG, Moulton MP, Selfridge RG (1998) Null matrices and the analysis of species co-occurrences. Oecologia 116: 275–283

Schluter D (1984) A variance test for detecting species associations, with some example applications. Ecology 65:998–1005

Shenk TM, White GC, Burnham KP (1998) Sampling-variance effects on detecting density dependence from temporal trends in natural populations. Ecol Monogr 68:445–463

Stone L, Roberts A (1990) The checkerboard score and species distributions. Oecologia 85:74–79

Weiher E, Keddy P (eds) (1999) Ecological assembly rules: perspectives, advances, retreats. Cambridge University Press, Cambridge

Wilson, JB (1987) Methods for detecting non-randomness in species co-occurrences: a contribution. Oecologia 73:579–582