The Annals of Statistics
2013, Vol. 41, No. 3, 1569–1592
DOI: 10.1214/13-AOS1131

EXACT SAMPLING AND COUNTING FOR FIXED-MARGIN MATRICES

BY JEFFREY W. MILLER[1,2] AND MATTHEW T. HARRISON[1]

Brown University

The uniform distribution on matrices with specified row and column sums is often a natural choice of null model when testing for structure in two-way tables (binary or nonnegative integer). Due to the difficulty of sampling from this distribution, many approximate methods have been developed. We will show that by exploiting certain symmetries, exact sampling and counting is in fact possible in many nontrivial real-world cases. We illustrate with real datasets including ecological co-occurrence matrices and contingency tables.

1. A motivating example. In ecology, co-occurrence tables are used to summarize biogeographical data. For instance, Table 1 indicates the presence/absence of 26 mammalian species in 28 mountain ranges in the American Southwest. When presented with such data, one might wonder: What factors control which species live in which habitats? In 1975, ecologist (and now, renowned author) Jared Diamond stunned the ecology community with the proposal of specific "assembly rules" governing the allocation of species to habitats. Diamond (1975) observed that certain pairs of species tended to occur together, and other pairs tended to be disjoint, suggesting that cooperation and competition play a key role. But did these patterns really reflect species interactions, or were they merely due to random chance?

To address this question, Connor and Simberloff (1979) suggested statistical hypothesis testing. Since some species are simply more prolific than others, and some habitats are larger than others, a sensible choice of null model is the uniform distribution on co-occurrence matrices with the observed numbers of habitats per species (row sums) and species per habitat (column sums). Connor and Simberloff (1979) presented a formidable challenge to Diamond's theory by showing that under this simple null model, the statistics observed by Diamond could easily have arisen by chance. A contentious debate erupted, yielding an extensive body of research on test statistics and null models for detecting various types of "structure" in ecological matrices. Decades later, the basic null model of Connor and Simberloff

Received January 2013; revised May 2013.
[1]Supported in part by NSF Grant DMS-10-07593 and DARPA contract FA8650-11-1-715.
[2]Supported by a NDSEG fellowship.
MSC2010 subject classifications. Primary 62H15; secondary 62H17, 05A15.
Key words and phrases. Exact sampling, exact counting, binary matrix, contingency table, integer points in polyhedra.

TABLE 1
26 *mammalian species in* 28 *mountain ranges* [*Patterson and Atmar* (1986)]

	Habitat																											
Species	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
B	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
C	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1
D	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	1	0	0	0	0
E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0	0
F	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	1	0	1	1	1	1	0	1	0	1	0	0	0
G	1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
H	1	1	1	1	1	1	0	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
I	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
J	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	1	1	1	1	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
M	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
N	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	1	1	0	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
X	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

has withstood the test of time, and is now a mainstay in the analysis of ecological matrices; see, for example, Gotelli and McCabe (2002), Ulrich and Gotelli (2007) and references therein.

As a concrete example, consider the montane mammals in Table 1. Patterson and Atmar (1986) proposed a model in which, during the most recent glacial period, cold-adapted species inhabited a region spanning several mountain ranges and the low-lying areas between, but in the current (warmer) interglacial period these populations have receded into the mountains and become extinct in some areas. They suggest that this would cause the set of species found in one mountain range to tend to be a subset of those found in another. This led them to consider the following *nested subset statistic*, equal to the number of species–habitat pairs such that the species does not occur in that habitat but does occur in a less-populated habitat,

$$S_{\text{nest}} = \sum_{i,j} I(a_{ij} = 0, q_j > m_i),$$

where $\mathbf{A} = (a_{ij})$ is a binary matrix with species as rows and habitats as columns (such as Table 1), $q_j = \sum_i a_{ij}$ and $m_i = \min\{q_j : a_{ij} = 1, j = 1, \ldots, n\}$. Here, $I(E)$ is 1 if E is true and 0 otherwise. (Note: Smaller S_{nest} means more "nestedness.") To perform a hypothesis test using the standard null model described above, one would estimate the p-value for S_{nest} by sampling from the uniform distribution over binary matrices with the observed row and column sums—in this case, (26, 26, 25, 22, 22, 18, 12, 12, 12, 11, 10, 10, 8, 8, 8, 7, 6, 6, 5, 5, 4, 4, 3, 3, 1, 1) and (26, 24, 23, 21, 19, 13, 13, 12, 11, 10, 10, 9, 9, 7, 7, 7, 7, 7, 7, 6, 6, 5, 5, 4, 3, 2, 1, 1), respectively. However, it is difficult to sample exactly from this distribution. Instead, Patterson and Atmar used an approximation in which the entries of each column are drawn proportionally to the row sums, conditioned on the column sum. (The row sums are not constrained in their approximation.) They drew 1000 samples from their approximation, estimated the p-value of S_{nest} to be 9×10^{-20} for Table 1, and concluded that the data does exhibit significantly more nestedness than one would expect under the null. Patterson and Atmar's (1986) article was highly influential, inspiring many subsequent studies into nestedness; see, for example, Ulrich and Gotelli (2007) and references therein.

The preceding scenario is commonplace—the combinatorial problem that arises from constraining the row and column sums makes it difficult to sample exactly from the desired uniform distribution. As a result, on all but the most trivial matrices, researchers have resorted to approximate methods, such as Markov chain Monte Carlo (MCMC), sequential importance sampling and heuristic approaches such as the one described above. With all these approximate methods, the nagging question remains: Did the use of an approximate distribution significantly affect the result?

In this work, we describe an efficient algorithm for sampling exactly from the uniform distribution over binary or nonnegative integer matrices with given row

TABLE 2
Sample statistics of S_{nest} for montane mammal data (Table 1)

Method	**# samples**	**estimated p-value**	**mean**	**std. dev.**	**min**	**max**
Exact	1,000,000	0.0322 ± 0.00018	80.7	9.7	44	132
Heuristic	1000	9×10^{-20}	227.9	18.1	180	287

and column sums (provided that most of the sums are not too large). As a result, Monte Carlo estimates of quantities of interest, such as p-values, can be accompanied by exact confidence intervals (i.e., true confidence intervals, rather than intervals based on asymptotic approximations). Further, our algorithm computes the exact number of such matrices.

In Table 2, we compare the results of Patterson and Atmar with results based on 10^6 exact samples using our algorithm. There are large discrepancies. We estimate the p-value to be 0.0322 ± 0.00018 ($\hat{p} \pm \hat{\text{se}}$), with an exact 95% confidence interval of $[0.0318, 0.0326]$ (based on the binomial c.d.f., not on $\hat{\text{se}}$). Their estimate of 9×10^{-20} is far smaller—in fact, the value of the test statistic on the observed matrix, $S_{\text{nest}} = 63$, is considerably less than the smallest value among all of their 1000 samples, $S_{\text{nest}} = 180$. (Note: It appears that they used a normal approximation to the distribution of S_{nest} to estimate the p-value.) Meanwhile, in view of the histogram of exact samples in Figure 1, the observed matrix appears relatively typical! Recall that in their approximation, the entries of each column are drawn

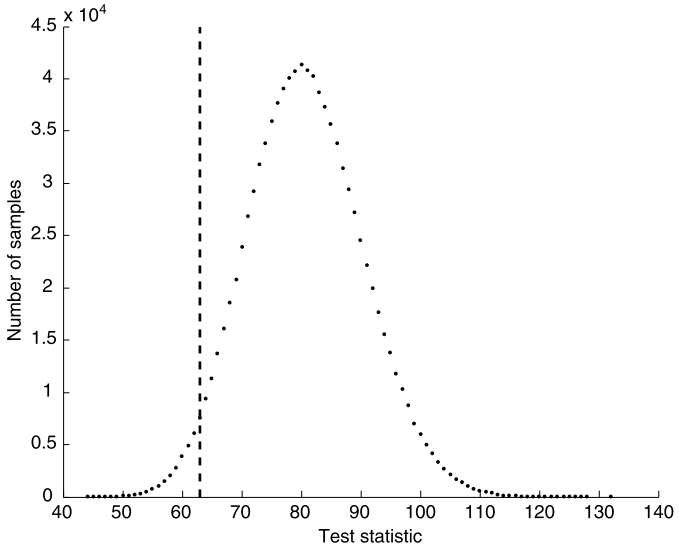

FIG. 1. *Histogram of S_{nest} for 10^6 exact samples from the uniform distribution over matrices with margins as in Table 1. The dashed line is the value of S_{nest} for Table 1.*

proportionally to the row sums, conditioned on the column sum, and that the row sums are not constrained. Apparently, omitting the constraint on the row sums has a drastic effect. As a result, the analysis dramatically underestimated the p-value. This vividly illustrates the utility of exact sampling in these problems.

Our algorithm required 46 seconds to find that there are 2,663,296,694,330,271, 332,856,672,902,543,209,853,700 ($\approx 2.7 \times 10^{39}$) binary matrices with row and column sums as in Table 1, and subsequently, required 4.2 milliseconds per exact sample. (All computations reported in this paper were performed using a 2.8 GHz processor with 6 GB of RAM.) We know of no other algorithm capable of exact counting and sampling for matrices of this size.

2. Overview. Let $N(\mathbf{p}, \mathbf{q})$ be the number of $m \times n$ binary matrices with margins (row and column sums) $\mathbf{p} = (p_1, \ldots, p_m) \in \mathbb{N}^m$ and $\mathbf{q} = (q_1, \ldots, q_n) \in \mathbb{N}^n$, respectively, and let $M(\mathbf{p}, \mathbf{q})$ be the corresponding number of $\mathbb{N}$-valued matrices. (We use $\mathbb{N} = \{0, 1, \ldots\}$ throughout.) In this paper we develop a technique for finding $N(\mathbf{p}, \mathbf{q})$ and $M(\mathbf{p}, \mathbf{q})$, and for exact uniform sampling from these sets of matrices—an important and challenging problem [Diaconis and Gangolli (1995), Chen et al. (2005)]. Our method is feasible for modestly-sized matrices (roughly, $m + n \leq 100$ with current desktop computing power) or very sparse large matrices.

As described above, in the binary case, an important application is testing for structure in ecological co-occurrence matrices. It turns out that most real-world ecology matrices are small enough that our method is feasible. In the $\mathbb{N}$-valued case, an important application is the conditional volume test of Diaconis and Efron (1985) for two-way contingency tables (for which our method is feasible as long as the margins are relatively small). In addition to these direct applications, a major auxiliary benefit of having an exact method is that it enables one to measure the accuracy of certain approximate methods (which can scale to matrices far larger than our exact method can accommodate).

Since a bipartite graph with degree sequences $\mathbf{p} = (p_1, \ldots, p_m) \in \mathbb{N}^m$, $\mathbf{q} = (q_1, \ldots, q_n) \in \mathbb{N}^n$ (and m, n vertices in each part, resp.) can be viewed as a $m \times n$ matrix with row and column sums $(\mathbf{p}, \mathbf{q})$, our technique applies equally well to counting and uniformly sampling such bipartite graphs. Under this correspondence, simple graphs correspond to binary matrices, and multigraphs correspond to $\mathbb{N}$-valued matrices.

The distinguishing characteristic of our method is its tractability on matrices of nontrivial size. In general, computing $M(\mathbf{p}, \mathbf{q})$ is #P-complete [Dyer, Kannan and Mount (1997)], and perhaps $N(\mathbf{p}, \mathbf{q})$ is as well. However, if one assumes a bound on the column sums then our algorithm computes both numbers in polynomial time. After counting, uniform samples may be drawn in polynomial expected time for bounded column sums. To our knowledge, all previous algorithms *for the non-regular case* require super-polynomial time (in the worst case) to compute these numbers, even for bounded column sums. (We assume a description length of at least $m + n$ and no more than $m \log a + n \log b$, where $a = \max p_i$, $b = \max q_i$.) In

general (without assuming a bound on the column sums), our algorithm computes $N(\mathbf{p}, \mathbf{q})$ or $M(\mathbf{p}, \mathbf{q})$ in $O(m(ab+c)(a+b)^{b-1}(b+c)^{b-1}(\log c)^3)$ time for $m \times n$ matrices, where $a = \max p_i$, $b = \max q_i$ and $c = \sum p_i = \sum q_i$. After counting, uniform samples may be drawn in $O(mc \log c)$ expected time.

In complement to most approaches to exactly computing $M(\mathbf{p}, \mathbf{q})$, which are efficient for small tables with large margins, our algorithm is efficient for large tables with small margins. For instance, computing $M(\mathbf{p}, \mathbf{q})$ for the 100×100 matrices with $\mathbf{p} = \mathbf{q} = (5^{(20)}, 4^{(20)}, 3^{(20)}, 2^{(20)}, 1^{(20)})$, where $x^{(k)}$ denotes x repeated k times, takes 701 seconds (the exact number, approximately $2.9580567 \times 10^{434}$, is available upon request). Likewise, computing $N(\mathbf{p}, \mathbf{q})$ for the same $\mathbf{p}$ and $\mathbf{q}$ takes 688 seconds (the number is approximately $2.3514766 \times 10^{431}$).

The remainder of the paper is organized as follows. In the rest of this section, we describe related work from the literature. In Section 3, we first describe the intuition behind our technique, then formally state the recursions and the resulting algorithm, and give bounds on computation time. In Section 4, we illustrate some applications. In Section 5 we prove the recursions and in Section 6, we prove the bounds on computation time.

2.1. *Previous work.* We briefly survey the previous work on this problem. This review is not exhaustive, focusing instead on those results which are particularly significant or closely related to the present work. Let $H_n(r)$ and $H_n^*(r)$ denote $M(\mathbf{p}, \mathbf{q})$ and $N(\mathbf{p}, \mathbf{q})$, respectively, when $\mathbf{p} = \mathbf{q} = (r, \ldots, r) \in \mathbb{N}^n$. The predominant focus has been on the regular cases $H_n(r)$ and $H_n^*(r)$.

Work on counting these matrices goes back at least as far as MacMahon [(1915), see Volume II, page 161], who applied his expansive theory to find the polynomial for $H_3(r)$. Redfield's theorem [Redfield (1927)], inspired by MacMahon, can be used to derive summations for some special cases, such as $H_n(r)$, $H_n^*(r)$ for $r = 2, 3$, and in similar work, Read (1959, 1960) used Pólya theory to derive these summations for $r = 3$. Two beautiful theoretical results must also be mentioned: Stanley (1973) proved that for fixed n, $H_n(r)$ is a polynomial in r, and Gessel (1987, 1990) showed that for fixed r, both $H_n(r)$ and $H_n^*(r)$ are P-recursive in n, vastly generalizing the recursions for $H_n(2)$, $H_n^*(2)$ found by Anand, Dumir and Gupta (1966).

We turn next to algorithmic results more closely related to the present work. McKay (1983) and Canfield and McKay (2005) have demonstrated a coefficient extraction technique for computing $N(\mathbf{p}, \mathbf{q})$ in the semi-regular case [in which $\mathbf{p} = (a, \ldots, a) \in \mathbb{N}^m$ and $\mathbf{q} = (b, \ldots, b) \in \mathbb{N}^n$]. To our knowledge, McKay's is the most efficient method known previously for $N(\mathbf{p}, \mathbf{q})$. By our analysis it requires at least $\Omega(mn^b)$ time for bounded a, b, while the method presented here is $O(mn^b(\log n)^3)$ in this case. Since this latter bound is quite crude, we expect that our method should have comparable or better performance, and indeed empirically we find that typically it is more efficient. If only b is bounded, McKay's algorithm is still $\Omega(mn^b)$, but the bound on our performance increases to $O(mn^{2b-1}(\log n)^3)$,

so it is possible that McKay's algorithm will outperform ours in these cases. Nonetheless, it is important to bear in mind that McKay's algorithm is efficient only in the semi-regular case (while our method permits nonregular margins). If neither a nor b is bounded, McKay's method is exponential in b (as is ours).

McKay and Wormald (1990) presented an intriguing randomized algorithm for exact uniform sampling from the set of binary matrices with margins $\mathbf{p}, \mathbf{q}$, provided that all of the margins are sufficiently small. It takes $O((m+n)^2k^4)$ expected time per sample, where k is an upper bound on all of the margins. Unfortunately, the conditions under which it applies are rather restrictive—in fact, for most of the real-world problems we have encountered, it reduces to a simple rejection sampling scheme that is rather inefficient.

Regarding $M(\mathbf{p}, \mathbf{q})$, one of the most efficient algorithms known to date is LattE (lattice point enumeration) [De Loera et al. (2004)], which uses the algorithm of Barvinok (1994) to count lattice points contained in convex polyhedra. It runs in polynomial time for any fixed dimension, and as a result it can compute $M(\mathbf{p}, \mathbf{q})$ for astoundingly large margins, provided that m and n are small. However, since the computation time grows very quickly with the dimension, LattE is currently inapplicable when m and n are larger than 6. There are similar algorithms [Mount (2000), De Loera and Sturmfels (2003), Beck and Pixton (2003)] that are efficient for small matrices.

In addition, several other algorithms have been presented for finding $N(\mathbf{p}, \mathbf{q})$ [such as Johnsen and Straume (1987), Pérez-Salvador et al. (2002), Wang (1988), Wang and Zhang (1998)] and $M(\mathbf{p}, \mathbf{q})$ [Diaconis and Gangolli (1995), Gail and Mantel (1977)] allowing nonregular margins; however, it appears that all are exponential in the size of the matrix, even for bounded margins. While in this work we are concerned solely with exact results, we note that many useful approximations for $N(\mathbf{p}, \mathbf{q})$ and $M(\mathbf{p}, \mathbf{q})$ (in the general case) have been found, as well as approximate sampling algorithms [Canfield, Greenhill and McKay (2008), Chen et al. (2005), Greenhill, McKay and Wang (2006), Harrison and Miller (2013), Holmes and Jones (1996)].

3. Main results.

3.1. *Idea of the algorithm.* Before formally presenting the results, we introduce the simple observation underlying our approach. The idea of recursively counting these matrices is old [Gail and Mantel (1977)]; the key novelty of our method is that it exploits symmetries arising from repeated column sums, leading to a dramatic improvement over naïve recursions. The basic idea is very straightforward—the surprising part is that it leads to an algorithm that is efficient for many nontrivial datasets.

For simplicity, consider the binary case. The first thing to notice is that $N(\mathbf{p}, \mathbf{q})$ is unchanged by permutations of the entries of $\mathbf{p}$ or $\mathbf{q}$ (as is easy to see). Now, suppose $\mathbf{p} = (6, 6, 4, 3, 1, 1, 1)$ and $\mathbf{q} = (0, 0, 1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 3)$,

TABLE 3
Partially-filled matrix

0	0	1	1	0	1	0	1	0	1	0	1	0	6
													6
													4
													3
													1
													1
													1
0	0	1	1	1	1	2	2	2	3	3	3	3	

and consider the partially-filled matrix in Table 3 in which the first row is $\mathbf{u} = (0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0)$ and the rest of the matrix is undetermined.

There are $N(\mathbf{p}, \mathbf{q})$ binary matrices with margins $(\mathbf{p}, \mathbf{q})$, and there are clearly $N(L\mathbf{p}, \mathbf{q} - \mathbf{u})$ of these with $\mathbf{u}$ as the first row, where L denotes the left-shift map: $L\mathbf{p} = (p_2, \ldots, p_m)$. Divide the first row into blocks $0, 1, 2, \ldots, m$ where block k contains the columns i such that $q_i = k$. Since the number of matrices is unchanged under permutations of the margins, then $N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) = N(L\mathbf{p}, \mathbf{q} - \mathbf{v})$ for any permutation $\mathbf{v}$ of $\mathbf{u}$ such that the number of ones in each block is unchanged. If r_k is the size of block k, and s_k is the number of ones in block k, then there are

$$\binom{\mathbf{r}}{\mathbf{s}} := \binom{r_1}{s_1} \cdots \binom{r_m}{s_m}$$

such permutations $\mathbf{v}$, where $\mathbf{r} = (r_1, \ldots, r_m)$ and $\mathbf{s} = (s_1, \ldots, s_m)$. For instance, in this example, $\binom{\mathbf{r}}{\mathbf{s}} = \binom{4}{3}\binom{3}{1}\binom{4}{2}$. Note that s_0 will be 0, and thus $\binom{r_0}{s_0}$ will be 1, for any $\mathbf{u}$ such that $N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) \neq 0$. Therefore,

$$N(\mathbf{p}, \mathbf{q}) = \sum_{\mathbf{u} \in \{0,1\}^n} N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) = \sum_{\mathbf{s}} \binom{\mathbf{r}}{\mathbf{s}} N(L\mathbf{p}, \mathbf{q} - \mathbf{u}_\mathbf{s}),$$

where the second sum is over nonnegative integer vectors $\mathbf{s} = (s_1, \ldots, s_m)$ summing to p_1, and $\mathbf{u}_\mathbf{s} \in \{0, 1\}^n$ is any binary vector with s_k ones in block k for each $k = 1, \ldots, m$, and zero ones in block 0. This defines a recursion for $N(\mathbf{p}, \mathbf{q})$ which, when carefully implemented using dynamic programming and the Gale–Ryser criterion (described below), is the basis for our algorithm. This computation yields a data structure that enables efficient sampling in a row-by-row fashion. There is a similar (although more subtle) recursion for the case of $M(\mathbf{p}, \mathbf{q})$.

3.2. *Recursions, algorithms and bounds.* Introducing the following notation will be useful. We consider $\mathbb{N}^n$ to be the subset of $\mathbb{N}^\infty := \{(r_1, r_2, \ldots) : r_i \in \mathbb{N} \text{ for } i = 1, 2, \ldots\}$ such that all but the first n components are zero. Let $L : \mathbb{N}^\infty \to \mathbb{N}^\infty$ denote the left-shift map: $L\mathbf{r} = (r_2, r_3, \ldots)$. Given $\mathbf{r}, \mathbf{s} \in \mathbb{N}^n$, let $\mathbf{r} \setminus \mathbf{s} := \mathbf{r} - \mathbf{s} + L\mathbf{s}$ (which may be read as "$\mathbf{r}$ *reduce* $\mathbf{s}$"), let $\binom{\mathbf{r}}{\mathbf{s}} := \binom{r_1}{s_1} \cdots \binom{r_n}{s_n}$, and let $\bar{\mathbf{r}}$ denote the vector of counts, $\bar{\mathbf{r}} := (\bar{r}_1, \bar{r}_2, \ldots)$ where $\bar{r}_i := \#\{j : r_j = i\}$. We write $\mathbf{r} \leq \mathbf{s}$

if $r_i \le s_i$ for all i. Given $n \in \mathbb{N}$, let $C_n(k) := \{\mathbf{r} \in \mathbb{N}^n : \sum_i r_i = k\}$ be the n-part compositions (including zeros) of k, and given $\mathbf{s} \in \mathbb{N}^n$, let $C^{\mathbf{s}}(k) := \{\mathbf{r} \in C_n(k) : \mathbf{r} \le \mathbf{s}\}$. Since $N(\mathbf{p}, \mathbf{q})$ and $M(\mathbf{p}, \mathbf{q})$ are fixed under permutations of the row sums $\mathbf{p}$ or column sums $\mathbf{q}$, and since zeros in the margins do not affect the number of matrices, then we may define $\bar{N}(\mathbf{p}, \bar{\mathbf{q}}) := N(\mathbf{p}, \mathbf{q})$ and $\bar{M}(\mathbf{p}, \bar{\mathbf{q}}) := M(\mathbf{p}, \mathbf{q})$ without ambiguity.

THEOREM 3.1 (Recursions). *The number of matrices with margins* $(\mathbf{p}, \mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$ *is given by*

$$(1)\quad \bar{N}(\mathbf{p}, \mathbf{r}) = \sum_{\mathbf{s} \in C^{\mathbf{r}}(p_1)} \binom{\mathbf{r}}{\mathbf{s}} \bar{N}(L\mathbf{p}, \mathbf{r} \setminus \mathbf{s}) \quad \textit{for binary matrices, and}$$

$$(2)\quad \bar{M}(\mathbf{p}, \mathbf{r}) = \sum_{\mathbf{s} \in C^{\mathbf{r}+L\mathbf{s}}(p_1)} \binom{\mathbf{r} + L\mathbf{s}}{\mathbf{s}} \bar{M}(L\mathbf{p}, \mathbf{r} \setminus \mathbf{s}) \quad \textit{for } \mathbb{N}\textit{-valued matrices,}$$

where $\mathbf{r} = \bar{\mathbf{q}}$, *and in* (2), *we sum over all* $\mathbf{s}$ *such that* $\mathbf{s} \in C^{\mathbf{r}+L\mathbf{s}}(p_1)$.

For the proof, see Section 5. In the binary case, computation of this sum is greatly simplified by summing only over those $\mathbf{s} \in C^{\mathbf{r}}(p_1)$ for which $\bar{N}(L\mathbf{p}, \mathbf{r} \setminus \mathbf{s})$ is nonzero. This can be efficiently achieved using the Gale–Ryser criterion [Gale (1957), Ryser (1957)], which provides the following necessary and sufficient condition for the existence of a binary matrix with margins $(\mathbf{p}, \mathbf{q})$: when $q'_i := \#\{j : q_j \ge i\}$ and $p_1 \ge \cdots \ge p_m$, we have $N(\mathbf{p}, \mathbf{q}) \ne 0$ if and only if $\sum_{i=1}^{j} p_i \le \sum_{i=1}^{j} q'_i$ for all $j < m$ and $\sum_{i=1}^{m} p_i = \sum_{i=1}^{m} q'_i$. This is easily translated into a condition in terms of $(\mathbf{p}, \bar{\mathbf{q}})$ and $\bar{N}(\mathbf{p}, \bar{\mathbf{q}})$. In the $\mathbb{N}$-valued case, there is no analogue to the Gale–Ryser criterion [since $M(\mathbf{p}, \mathbf{q}) > 0$ for any nonnegative $\mathbf{p}, \mathbf{q}$ such that $\sum_i p_i = \sum_i q_i$]. The following recursive procedure can be used to compute either $N(\mathbf{p}, \mathbf{q})$ or $M(\mathbf{p}, \mathbf{q})$.

ALGORITHM 3.2 (Counting).
Input: $(\mathbf{p}, \bar{\mathbf{q}})$, *where* $(\mathbf{p}, \mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$ *are margins such that* $\sum_i p_i = \sum_i q_i$.
Output: $N(\mathbf{p}, \mathbf{q})$ [*or* $M(\mathbf{p}, \mathbf{q})$], *the number of binary* (*or* $\mathbb{N}$*-valued*) *matrices*.
Storage: *Lookup table initialized with* $\bar{N}(\mathbf{0}, \mathbf{0}) = 1$ [*or* $\bar{M}(\mathbf{0}, \mathbf{0}) = 1$].

(1) *If* $\bar{N}(\mathbf{p}, \bar{\mathbf{q}})$ [*or* $\bar{M}(\mathbf{p}, \bar{\mathbf{q}})$] *is in the lookup table, return the result.*
(2) *In the binary case, if Gale–Ryser gives* $\bar{N}(\mathbf{p}, \bar{\mathbf{q}}) = 0$, *store the result and return* 0.
(3) *Evaluate the sum in Theorem* 3.1, *recursing to step* (1) *for each term.*
(4) *Store the result and return it.*

Let $T(\mathbf{p}, \mathbf{q})$ be the time (number of machine operations) required by Algorithm 3.2 to compute $N(\mathbf{p}, \mathbf{q})$ or $M(\mathbf{p}, \mathbf{q})$, after performing an $O(n^3)$ preprocessing step to compute all needed binomial coefficients. [It turns out that computing

$M(\mathbf{p},\mathbf{q})$ always takes longer, but the bounds we prove apply to both $N(\mathbf{p},\mathbf{q})$ and $M(\mathbf{p},\mathbf{q})$.] We give a series of bounds on $T(\mathbf{p},\mathbf{q})$ ranging from tighter but more complicated, to more crude but simpler. The bounds will absorb the $O(n^3)$ precomputation except in the trivial case when the maximum column sum is 1.

THEOREM 3.3 (Bounds). *Suppose* $(\mathbf{p},\mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$, $a = \max p_i$, $b = \max q_i$ *and* $c = \sum p_i = \sum q_i$. *Then*:

(1) $T(\mathbf{p},\mathbf{q}) = O\big((ab+c)(\log c)^3 \sum_{i=1}^{m} \binom{p_i+b-1}{b-1}\binom{p_i+\cdots+p_m+b-1}{b-1}\big)$,
(2) $T(\mathbf{p},\mathbf{q}) = O(m(ab+c)(a+b)^{b-1}(b+c)^{b-1}(\log c)^3)$,
(3) $T(\mathbf{p},\mathbf{q}) = O(mn^{2b-1}(\log n)^3)$ *for bounded* b,
(4) $T(\mathbf{p},\mathbf{q}) = O(mn^b(\log n)^3)$ *for bounded* a, b.

For the proof, see Section 6. Since we may swap $\mathbf{p}$ and $\mathbf{q}$ without changing the number of matrices, we could use Algorithm 3.2 on $(\mathbf{q},\bar{\mathbf{p}})$ to compute $N(\mathbf{p},\mathbf{q})$ or $M(\mathbf{p},\mathbf{q})$ using $T(\mathbf{q},\mathbf{p})$ operations, which, for example, is $O(nm^a(\log m)^3)$ for bounded a, b. Typically, we find that choosing $\mathbf{p}$ to be the shorter of the two vectors is preferable. $T(\mathbf{p},\mathbf{q})$ also depends on the ordering of the row sums $p_1,\ldots,p_m$ as suggested by Theorem 3.3(1), and we find that putting them in decreasing order $p_1 \geq \cdots \geq p_m$ tends to work well. In the binary case, Algorithm 3.2 is typically made significantly more efficient by using the Gale–Ryser conditions, and this is not accounted for in these bounds.

It is worth mentioning that a significant further reduction in computation time can be achieved by factoring the sums in Theorem 3.1. For example, in the binary case, we use

$$\bar{N}(\mathbf{p},\mathbf{r}) = \sum_{s_1}\binom{r_1}{s_1}\sum_{s_2}\binom{r_2}{s_2}\cdots\sum_{s_m}\binom{r_m}{s_m}\bar{N}(L\mathbf{p},\mathbf{r}\setminus\mathbf{s}),$$

where for each $k = 1,\ldots,m$, s_k is summed over a range of values chosen so that $\mathbf{s}$ will always satisfy both $\mathbf{s} \in C^{\mathbf{r}}(p_1)$ and the Gale–Ryser criterion. This improvement is also not accounted for in the bounds above.

Algorithm 3.2 traverses a directed acyclic graph in which each node represents a distinct set of input arguments $(\mathbf{p},\bar{\mathbf{q}})$ to the algorithm. Node $(\mathbf{u},\bar{\mathbf{v}})$ is the child of node $(\mathbf{p},\bar{\mathbf{q}})$ if the algorithm is called (recursively) with arguments $(\mathbf{u},\bar{\mathbf{v}})$ while executing a call with arguments $(\mathbf{p},\bar{\mathbf{q}})$. We associate with each node its *count*: the number of matrices with the corresponding margins. If the initial input arguments are $(\mathbf{p},\bar{\mathbf{q}})$, then all nodes are descendants of node $(\mathbf{p},\bar{\mathbf{q}})$. Meanwhile, all nodes with positive count are ancestors of node $(\mathbf{0},\mathbf{0})$. Note the correspondence between the children of a node $(\mathbf{u},\bar{\mathbf{v}})$ and the compositions $\mathbf{s} \in C^{\bar{\mathbf{v}}}(u_1)$ in the binary case, and $\mathbf{s} \in C^{\bar{\mathbf{v}}+L\mathbf{s}}(u_1)$ in the $\mathbb{N}$-valued case, under which $\mathbf{s}$ corresponds with the child $(L\mathbf{u},\bar{\mathbf{v}}\setminus\mathbf{s})$.

Once the counting is complete, these counts yield an efficient algorithm for uniform sampling from the set of $(\mathbf{p},\mathbf{q})$ matrices (binary or $\mathbb{N}$-valued). It is straightforward to see that since the counts are exact, the following algorithm yields a sample from the uniform distribution.

ALGORITHM 3.4 (Sampling).
Input:
• *Row and column sums* $(\mathbf{p}, \mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$ *such that* $\sum_i p_i = \sum_i q_i$.
• *Lookup table of counts generated by Algorithm 3.2 on input* $(\mathbf{p}, \bar{\mathbf{q}})$.
Output: *A uniformly-drawn binary* (*or* $\mathbb{N}$*-valued*) *matrix with margins* $(\mathbf{p}, \mathbf{q})$.

(1) *Initialize* $(\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{p}, \mathbf{q})$.
(2) *If* $(\mathbf{u}, \mathbf{v}) = (\mathbf{0}, \mathbf{0})$, *exit*.
(3) *Choose a child* $(L\mathbf{u}, \bar{\mathbf{v}} \setminus \mathbf{s})$ *of* $(\mathbf{u}, \bar{\mathbf{v}})$ *with probability proportional to its count times the number of corresponding rows* [*i.e.*, *the number of rows* $\mathbf{r} \in C^{\mathbf{v}}(u_1)$ *such that* $\overline{\mathbf{v} - \mathbf{r}} = \bar{\mathbf{v}} \setminus \mathbf{s}$].
(4) *Choose a row* $\mathbf{r}$ *uniformly among the corresponding rows*.
(5) $(\mathbf{u}, \mathbf{v}) \leftarrow (L\mathbf{u}, \mathbf{v} - \mathbf{r})$.
(6) *Go to* (2).

In step (3), there are $\binom{\bar{\mathbf{v}}}{\mathbf{s}}$ corresponding rows $\mathbf{r}$ in the binary case (in which $\mathbf{r} \in \{0, 1\}^n$), and $\binom{\bar{\mathbf{v}}+L\mathbf{s}}{\mathbf{s}}$ in the $\mathbb{N}$-valued case. In Section 6, we prove that Algorithm 3.4 takes $O(mc \log c)$ expected time per sample, where $c = \sum_i p_i$.

A software implementation of the algorithms above has been made available, and contains demonstrations of the applications in Section 1 above and Section 4 below.

4. Applications.

4.1. *Zero–one tables*. In several fields of study, including neurophysiology (multivariate binary time series), sociology (affiliation matrices), psychometrics (item response theory) and ecology (co-occurrence matrices), the uniform distribution on binary matrices with fixed margins is applicable, particularly in the context of conditional inference. In this section, we take a closer look at the ecology application, continuing the discussion from Section 1. It turns out that many real-world ecology datasets can be accommodated by our method: out of 291 ecology matrices in a collection compiled by Atmar and Patterson (1995), it could handle 225.

In Section 1, we compared our approach to a heuristic approximation. However, it is now more common for researchers to use MCMC [e.g., Gotelli and McCabe (2002), Ulrich and Gotelli (2007)], and recently, sequential importance sampling (SIS) approaches have been developed that appear to improve upon MCMC [Chen et al. (2005), Harrison and Miller (2013)].

To compare with these alternatives, we consider a benchmark dataset for this application. Table 4 indicates the presence/absence of 13 species of finch on 17 of the Galápagos Islands. It comes equipped with the colorful name "Darwin's finches" because Charles Darwin's development of the theory of evolution was inspired in part by his observations of these birds. The row and column sums of the matrix are (14, 13, 14, 10, 12, 2, 10, 1, 10, 11, 6, 2, 17) and (4, 4, 11, 10,

TABLE 4
13 *species of finch in* 17 *of the Galápagos Islands* [*Chen et al.* (2005)]

	Habitat																
Species	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
A	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
C	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0
D	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1	1	1
E	1	1	1	0	1	1	1	1	1	1	0	1	0	1	1	0	0
F	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
G	0	0	1	1	1	1	1	1	1	0	0	1	0	1	1	0	0
H	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
I	0	0	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
J	0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
K	0	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0	0
L	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
M	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

10, 8, 9, 10, 8, 9, 3, 10, 4, 7, 9, 3, 3), respectively. This table is the subject of an analysis by Chen et al. (2005), in which they use MCMC and their SIS algorithm to estimate the p-value for the $\overline{S^2}$ statistic of Roberts and Stone (1990),

$$\overline{S^2} = \frac{1}{\binom{m}{2}} \sum_{i<j} s_{ij}^2,$$

where $\mathbf{S} = (s_{ij}) = \mathbf{A}\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}$ is an $m \times n$ co-occurrence matrix. For given margins, larger values of $\overline{S^2}$ are interpreted as indicating greater competition and/or cooperation among species; see the Appendix for details.

The results of Chen et al. are reported in Table 5, alongside our results using exact sampling. Our results largely confirm the conclusion of Chen et al.—the p-value is small, leading one to reject the null hypothesis; see Figure 2. In the Appendix, we examine the sensitivity of this p-value to possible data-collection errors and to the choice of test statistic.

TABLE 5
Sample statistics of $\overline{S^2}$ *for Darwin's finch data* (*Table 4*)

Method	**# samples**	**Estimated p-value**
Exact	1,000,000	$(4.67 \pm 0.22) \times 10^{-4}$
SIS	1,000,000	$(3.96 \pm 0.36) \times 10^{-4}$
MCMC	15,000,000	$(3.56 \pm 0.68) \times 10^{-4}$

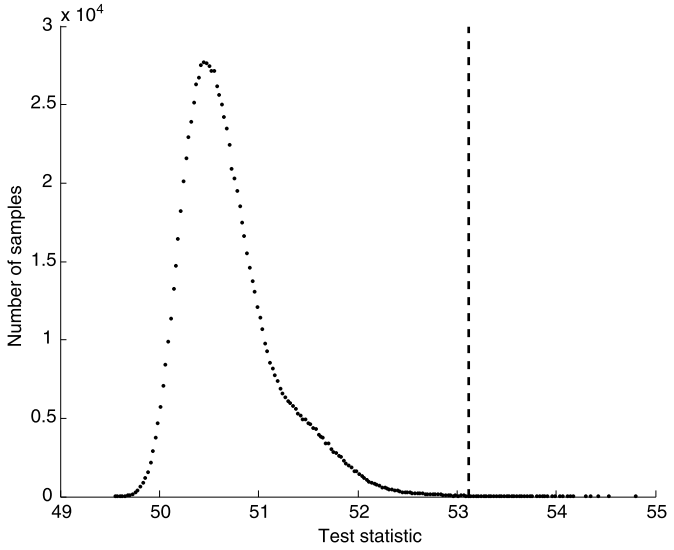

FIG. 2. *Histogram of* $\overline{S^2}$ *for* 10^6 *exact samples from the uniform distribution over matrices with margins as in Table* 4. *The dashed line is the value of* $\overline{S^2}$ *for Table* 4.

The computation time for our method is very competitive: for this dataset, counting the number of matrices takes 0.02 seconds, and exact sampling takes 0.16 milliseconds/sample, compared to 1.1 milliseconds/sample for SIS and 0.07 milliseconds/sample for MCMC. (However, on large matrices, MCMC and SIS should be significantly faster than our method, due to the overhead incurred by counting.) The SIS algorithm of Chen et al. also yields an estimate of 6.7150×10^{16} for the number of matrices. We compute the exact number of matrices to be 67,149,106,137,567,626, and note that this agrees with the exact number reported by Chen et al. (which they obtained by other means).

We should emphasize, however, that in comparison with MCMC and SIS, the appeal of our method is not its speed, but rather its exactness. There are no guarantees that the estimates coming from MCMC or SIS have adequately converged. Meanwhile, we are drawing exact i.i.d. samples from the null distribution, which provides many guarantees. For example, using the 10^6 exact samples above, we obtain an exact 95% confidence interval of $[4.26, 5.12] \times 10^{-4}$ for the p-value. [By "exact," we mean that it is a true confidence interval, with no approximations involved. Reported confidence intervals often involve two approximations: (1) approximate samples used for estimation, and/or (2) approximate interval construction based on asymptotics, such as the normal approximation to the binomial.] In a longer run of 10^9 exact samples, we estimate the p-value to be 4.672×10^{-4} and obtain an exact 95% confidence interval of $[4.66, 4.69] \times 10^{-4}$ (or 99.9% interval of $[4.65, 4.70] \times 10^{-4}$). Interestingly, this only barely overlaps with the SIS

TABLE 6
Heights of 205 *married couples* [*Galton* (1889)], *and two related tables*

(a)

	s	m	t
t	12	20	18
m	25	51	28
s	9	28	14

(b)

8	14	28
20	61	23
18	24	9

(c)

16	28	56
40	122	46
36	48	18

approximate 95% confidence interval, $[3.25, 4.67] \times 10^{-4}$. Clearly, it is advantageous to use the exact method.

4.2. *Contingency tables*. Two-way contingency tables are a large class of $\mathbb{N}$-valued matrices that arise frequently in statistics. Pearson's chi-square is the classical test of independence in such tables, however, when the independence hypothesis fails, it can be misleading to interpret the chi-square p-value as a measure of the deviation from independence. As a starting point for quantifying the departure from independence, Diaconis and Efron (1985) propose the conditional volume test (CVT): place the uniform distribution over tables with the observed margins, and compute the probability of observing a chi-square value less than that of the observed table (in other words, compute the p-value under this new null distribution). Larger values of the CVT p-value indicate greater deviation from independence.

Since our method enables exact sampling from the CVT null distribution, it can be used to estimate the p-value for the CVT. While many contingency tables that arise in practice will have margins that are too large for our method, some will fall within the feasible range.

To illustrate, Table 6(a) shows Francis Galton's (1889) data recording the heights (s = short, m = medium, t = tall) of 205 married couples (e.g., medium–short, tall–tall, etc.). Table 6(b) has the same margins but different entries, and in Table 6(c), every entry is exactly double that of Table 6(b). For (a), (b) and (c), Table 7 shows the results of Pearson's chi-square test (the approximate p-value)

TABLE 7
Pearson's chi-square and the conditional volume test of Diaconis and Efron (1985)

	Chi-square p-value	**CVT p-value**	**CVT 95% CI**
Table 6(a)	0.57	0.0011	[0.0005, 0.0020]
Table 6(b)	1.2×10^{-5}	0.13	[0.121, 0.136]
Table 6(c)	1.8×10^{-11}	0.13	[0.123, 0.137]

and the conditional volume test (the estimated p-value and exact 95% confidence interval).

The chi-square test and the CVT both indicate that (a) is close to independence, and that (b) is not close to independence. For (c), a naïve interpretation of the chi-square p-value would be that (c) is far further from independence than (b); meanwhile, the CVT indicates that it deviates from independence by roughly the same amount as (b), as one would expect.

These CVT results were each obtained using 10^4 exact samples, drawn at a rate of 0.14 seconds/sample for Tables 6(a) and (b) (after 0.3 seconds required to count the 1,268,792 tables), and 1.94 seconds/sample for Table 6(c) (after 7.7 seconds required to count the 19,151,218 tables).

We test our method further with two examples from Diaconis and Gangolli (1995). The first is an artificial 5×3 table, with margins $(10, 62, 13, 11, 39)$, $(65, 25, 45)$. Our algorithm takes 2.3 seconds to count the 239,382,173 corresponding tables, and 0.3 seconds/sample. Their second example is significantly more challenging: a 4×4 table recording the eye color and hair color of 592 subjects, with margins $(220, 215, 93, 64)$, $(108, 286, 71, 127)$. Our algorithm takes 16,145 seconds to count the 1,225,914,276,768,514 corresponding tables, and 90 seconds/sample. These counting results match the exact numbers reported by Diaconis and Gangolli (1995), obtained by other means.

Our method is not well-suited to small tables with large margins (like the last example), since it exploits the symmetries that arise when there are many columns. For small tables, there are significantly faster algorithms for counting, such as LattE [De Loera et al. (2004)]. In particular, these algorithms can handle extremely large margins (while ours cannot). However, such algorithms do not scale well to larger tables. In contrast, our method can handle somewhat larger tables, as long as the margins are sufficiently small—consider, for example, the 100×100 example at the end of Section 2.

5. Proof of recursions. In this section, we prove Theorem 3.1, making rigorous the intuitive argument given in Section 3.1. As an alternative to the "direct" proof below, in Miller and Harrison (2011) we also provide a generating function proof that employs some of the beautiful properties of symmetric functions and yields results of a more general nature.

For $\mathbf{r}, \mathbf{s} \in \mathbb{N}^\infty$, let $\mathbf{r} \wedge \mathbf{s}$ denote the component-wise minimum, that is, $(r_1 \wedge s_1, r_2 \wedge s_2, \ldots)$. In particular, $\mathbf{r} \wedge \mathbf{1} = (r_1 \wedge 1, r_2 \wedge 1, \ldots)$. Recall our convention that $\mathbb{N}^n$ is considered to be the subset of $\mathbb{N}^\infty$ such that all but the first n components are zero.

PROOF OF THEOREM 3.1. (1) Consider the binary case. Let $(\mathbf{p}, \mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$. If $\mathbf{r} = \bar{\mathbf{q}}$, then

$$\begin{aligned}\bar{N}(\mathbf{p}, \mathbf{r}) = N(\mathbf{p}, \mathbf{q}) &\overset{(a)}{=} \sum_{\mathbf{u} \in C^{\mathbf{q}\wedge\mathbf{1}}(p_1)} N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) \\ &\overset{(b)}{=} \sum_{\substack{\mathbf{s} \in \mathbb{N}^\infty : \\ \sum s_i = p_1}} \sum_{\substack{\mathbf{u} \in C^{\mathbf{q}\wedge\mathbf{1}}(p_1) : \\ \overline{\mathbf{q}-\mathbf{u}} = \bar{\mathbf{q}} \setminus \mathbf{s}}} N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) \\ &\overset{(c)}{=} \sum_{\mathbf{s} \in C^{\mathbf{r}}(p_1)} \binom{\mathbf{r}}{\mathbf{s}} \bar{N}(L\mathbf{p}, \mathbf{r} \setminus \mathbf{s}).\end{aligned}$$

Step (a) follows from partitioning the set of $(\mathbf{p}, \mathbf{q})$ matrices according to the first row $\mathbf{u} \in C^{\mathbf{q}\wedge\mathbf{1}}(p_1)$ of the matrix. Now, it is straightforward to check that for any $\mathbf{u} \in C^{\mathbf{q}}(p_1)$ there is a unique $\mathbf{s} \in \mathbb{N}^\infty$ such that $\overline{\mathbf{q} - \mathbf{u}} = \bar{\mathbf{q}} \setminus \mathbf{s}$ and $\sum s_i < \infty$, namely $s_i = \sum_{j=i}^{\infty} (\bar{q}_j - (\overline{\mathbf{q} - \mathbf{u}})_j)$, and it follows that $\sum s_i = p_1$. Step (b) partitions $C^{\mathbf{q}\wedge\mathbf{1}}(p_1)$ into the level sets of the map taking $\mathbf{u}$ to this $\mathbf{s} \in \mathbb{N}^\infty$. Step (c) follows since if $\overline{\mathbf{q} - \mathbf{u}} = \bar{\mathbf{q}} \setminus \mathbf{s}$ and $\mathbf{r} = \bar{\mathbf{q}}$, then $N(L\mathbf{p}, \mathbf{q} - \mathbf{u}) = \bar{N}(L\mathbf{p}, \overline{\mathbf{q} - \mathbf{u}}) = \bar{N}(L\mathbf{p}, \mathbf{r} \setminus \mathbf{s})$, and by Lemma 5.1(1) (below) the inner sum contains $\binom{\mathbf{r}}{\mathbf{s}}$ terms. The range of the sum reduces to $C^{\mathbf{r}}(p_1)$ since $\binom{\mathbf{r}}{\mathbf{s}}$ is zero unless $\mathbf{r} \le \mathbf{s}$.

(2) The $\mathbb{N}$-valued case is nearly identical, with the obvious changes [replace $C^{\mathbf{q}\wedge\mathbf{1}}(p_1)$, $C^{\mathbf{r}}(p_1)$ and $\binom{\mathbf{r}}{\mathbf{s}}$ by $C^{\mathbf{q}}(p_1)$, $C^{\mathbf{r}+L\mathbf{s}}(p_1)$ and $\binom{\mathbf{r}+L\mathbf{s}}{\mathbf{s}}$, resp., and use Lemma 5.1(2)]. □

LEMMA 5.1. *If* $\mathbf{v} \in \mathbb{N}^n$, $\mathbf{s} \in \mathbb{N}^\infty$ *and* $k = \sum s_i < \infty$, *then*:

(1) $\#\{\mathbf{u} \in C^{\mathbf{v}\wedge\mathbf{1}}(k) : \overline{\mathbf{v} - \mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}\} = \binom{\bar{\mathbf{v}}}{\mathbf{s}}$ *and*

(2) $\#\{\mathbf{u} \in C^{\mathbf{v}}(k) : \overline{\mathbf{v} - \mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}\} = \binom{\bar{\mathbf{v}}+L\mathbf{s}}{\mathbf{s}}$.

PROOF. (1) Let $b = \max v_i$. If $\mathbf{u} \in C^{\mathbf{v}\wedge\mathbf{1}}(k)$ and $\overline{\mathbf{v} - \mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$, then $s_i = 0$ for $i > b$, and for $i = b, b-1, \ldots, 1$ (starting with $i = b$ and working our way down), the fact that $(\overline{\mathbf{v} - \mathbf{u}})_i = \bar{v}_i - s_i + s_{i+1}$ implies that out of the $\bar{v}_i$ places j such that $v_j = i$, a subset of exactly s_i of them has $u_j = 1$. On the other hand, for any such sequence of subsets, there is a unique $\mathbf{u} \in C^{\mathbf{v}\wedge\mathbf{1}}(k)$ satisfying $\overline{\mathbf{v} - \mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$ which gives rise to it in this way. Clearly, there are $\binom{\bar{\mathbf{v}}}{\mathbf{s}}$ such sequences of subsets.

(2) We may as well assume $\mathbf{s} \le \bar{\mathbf{v}} + L\mathbf{s}$, since otherwise both sides are trivially zero. Let $b = \max v_i$, and note that $s_i = 0$ for all $i > b$. Consider the following algorithm for choosing a $\mathbf{u} \in C^{\mathbf{v}}(k)$ such that $\overline{\mathbf{v} - \mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$. Initialize $\mathbf{u}$ to be identically zero. There are $\bar{v}_b$ places j where $v_j = b$, so in order to satisfy $(\overline{\mathbf{v} - \mathbf{u}})_b = \bar{v}_b - s_b$, increment $u_j \leftarrow u_j + 1$ for s_b of these places (and keep the other $\bar{v}_b - s_b$ unchanged). Then after this, there are $\bar{v}_{b-1} + s_b$ places j where $v_j - u_j = b - 1$ (namely, the $\bar{v}_{b-1}$ places where $v_j = b - 1$, along with the s_b places where $v_j = b$ and $u_j = 1$ from the previous step), so to satisfy

$(\overline{\mathbf{v}-\mathbf{u}})_{b-1} = \bar{v}_{b-1} - s_{b-1} + s_b$, increment $u_j \leftarrow u_j + 1$ for s_{b-1} of these places. Continuing in this way, for $i = b-2, \ldots, 1$: there are $\bar{v}_i + s_{i+1}$ places j where $v_j - u_j = i$ (the $\bar{v}_i$ places where $v_j = i$, along with the s_{i+1} places where $v_j > i$ and $u_j = v_j - i$ due to previous steps), so to satisfy $(\overline{\mathbf{v}-\mathbf{u}})_i = \bar{v}_i - s_i + s_{i+1}$, increment $u_j \leftarrow u_j + 1$ for s_i of these places.

Clearly, there are

$$\binom{\bar{v}_b}{s_b}\binom{\bar{v}_{b-1}+s_b}{s_{b-1}}\cdots\binom{\bar{v}_1+s_2}{s_1} = \binom{\bar{\mathbf{v}}+L\mathbf{s}}{\mathbf{s}}$$

ways to follow this algorithm. By construction, any $\mathbf{u}$ obtained via the algorithm satisfies $(\overline{\mathbf{v}-\mathbf{u}})_i = \bar{v}_i - s_i + s_{i+1}$ for each $i = 1, \ldots, b$, $\mathbf{0} \le \mathbf{u} \le \mathbf{v}$ and $\sum_{j=1}^n u_j = \sum_{i=1}^b s_i$, and therefore $\mathbf{u} \in C^{\mathbf{v}}(k)$ and $\overline{\mathbf{v}-\mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$. Meanwhile, given any $\mathbf{u} \in C^{\mathbf{v}}(k)$ such that $\overline{\mathbf{v}-\mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$, there is a unique way to obtain it via the algorithm, since for $i = b, \ldots, 1$, the choice of which $\bar{v}_i - s_i + s_{i+1}$ places to leave unchanged is uniquely determined. Therefore, the number of ways to follow the algorithm equals the number of $\mathbf{u} \in C^{\mathbf{v}}(k)$ such that $\overline{\mathbf{v}-\mathbf{u}} = \bar{\mathbf{v}} \setminus \mathbf{s}$. □

6. Computation time. Let $W(\mathbf{r}) := \sum_{k=1}^n k r_k$ = the *weight* of $\mathbf{r} \in \mathbb{Z}^n$.

LEMMA 6.1 (Properties of the weight). *If* $\mathbf{r}, \mathbf{s} \in \mathbb{Z}^n$, *then*:

(1) $W(\mathbf{r}+\mathbf{s}) = W(\mathbf{r}) + W(\mathbf{s})$;
(2) $W(\mathbf{s} - L\mathbf{s}) = \sum s_i$;
(3) $W(\mathbf{r} \setminus \mathbf{s}) = W(\mathbf{r}) - \sum s_i$;
(4) $W(\bar{\mathbf{s}}) = \sum s_i$.

PROOF. All four are simple calculations. □

For the rest of this section, fix $(\mathbf{p}, \mathbf{q}) \in \mathbb{N}^m \times \mathbb{N}^n$ such that $\sum p_i = \sum q_i$, and consider $(\mathbf{p}, \mathbf{q})$ to be the margins of a set of $m \times n$ matrices. First, we address the time to compute $N(\mathbf{p}, \mathbf{q})$ using Algorithm 3.2, and $M(\mathbf{p}, \mathbf{q})$ will follow easily.

Let $\mathcal{D}(\mathbf{p}, \mathbf{q})$ denote the set of nontrivial nodes $(\mathbf{u}, \bar{\mathbf{v}})$ in the directed acyclic graph (as discussed in Section 3) descending from $(\mathbf{p}, \bar{\mathbf{q}})$ [including $(\mathbf{p}, \bar{\mathbf{q}})$], where nontrivial means $(\mathbf{u}, \bar{\mathbf{v}}) \ne (\mathbf{0}, \mathbf{0})$. Let $\Delta_k(j) := \{\mathbf{s} \in \mathbb{N}^k : W(\mathbf{s}) = j\}$ for $j, k \in \mathbb{N}$. The intuitive content of the following lemma is that the graph descending from $(\mathbf{p}, \bar{\mathbf{q}})$ is contained in a union of sets $\Delta_k(j)$ with weights decreasing by steps of $p_1, \ldots, p_m$.

LEMMA 6.2 (Descendants). *If* $t_j = \sum_{i=j}^m p_i$ *and* $b = \max q_i$, *then*

$$\mathcal{D}(\mathbf{p}, \mathbf{q}) \subset \{(\mathbf{u}, \bar{\mathbf{v}}) : \mathbf{u} = L^{j-1}\mathbf{p}, \bar{\mathbf{v}} \in \Delta_b(t_j),\ j = 1, \ldots, m\}.$$

PROOF. By the form of the recursion, $(\mathbf{u}, \bar{\mathbf{v}}) \in \mathcal{D}(\mathbf{p}, \mathbf{q})$ if and only if for some $1 \le j \le m$ there exist $\mathbf{s}^1, \dots, \mathbf{s}^{j-1}$ in $C^{\mathbf{r}^1}(p_1), \dots, C^{\mathbf{r}^{j-1}}(p_{j-1})$, respectively, with $\mathbf{r}^1 = \bar{\mathbf{q}}$, $\mathbf{r}^{i+1} = \mathbf{r}^i \setminus \mathbf{s}^i$ for $i = 1, \dots, j-1$, such that $(\mathbf{u}, \bar{\mathbf{v}}) = (L^{j-1}\mathbf{p}, \mathbf{r}^j)$. For $j \ge 2$, by Lemma 6.1 (3 and 4),

$$\begin{aligned} W(\mathbf{r}^j) &= W(\mathbf{r}^{j-1} \setminus \mathbf{s}^{j-1}) = W(\mathbf{r}^{j-1}) - p_{j-1} = W(\mathbf{r}^{j-2}) - p_{j-2} - p_{j-1} \\ &= \cdots = W(\mathbf{r}^1) - (p_1 + \cdots + p_{j-1}) \\ &= \sum_{i=1}^{n} q_i - \sum_{i=1}^{j-1} p_i = \sum_{i=1}^{m} p_i - \sum_{i=1}^{j-1} p_i = t_j \end{aligned}$$

and $\mathbf{r}^j \in \mathbb{N}^b$ by construction, so $\mathbf{r}^j \in \Delta_b(t_j)$. Hence, $(\mathbf{u}, \bar{\mathbf{v}}) = (L^{j-1}\mathbf{p}, \mathbf{r}^j)$ belongs to the set as claimed. □

Let $T(\mathbf{p}, \mathbf{q})$ be the time (number of machine operations) required by the algorithm (Algorithm 3.2) to compute $N(\mathbf{p}, \mathbf{q})$ after precomputing all needed binomial coefficients. Let $\tau(\mathbf{u}, \bar{\mathbf{v}})$ be the time to compute $\bar{N}(\mathbf{u}, \bar{\mathbf{v}})$ given $\bar{N}(L\mathbf{u}, \bar{\mathbf{v}} \setminus \mathbf{s})$ for all $\mathbf{s} \in C^{\bar{\mathbf{v}}}(u_1)$. That is, $T(\mathbf{p}, \mathbf{q})$ is the time to perform the entire recursive computation, whereas $\tau(\mathbf{u}, \bar{\mathbf{v}})$ is the time to perform a given call to the algorithm not including time spent in subcalls.

Let $n_0 := \#\{i : q_i > 0\}$ denote the number of nonempty columns. By constructing Pascal's triangle, we precompute all of the binomial coefficients that may be needed, and store them in a lookup table. We only need binomial coefficients with entries less or equal to n_0, for the following reason. In the binary case, the recursion involves numbers of the form $\binom{\bar{\mathbf{v}}}{\mathbf{s}}$ with $\mathbf{s} \le \bar{\mathbf{v}}$, and for any descendent $(\mathbf{u}, \bar{\mathbf{v}})$ and any $i > 0$ we have $\bar{v}_i \le n_0$ since the number of columns with sum i is less or equal to the total number of nonempty columns. For the $\mathbb{N}$-valued case, the same set of binomial coefficients will be sufficient, since then we have numbers of the form $\binom{\bar{\mathbf{v}}+L\mathbf{s}}{\mathbf{s}}$ with $\mathbf{s} \le \bar{\mathbf{v}} + L\mathbf{s}$, and thus

$$\bar{v}_i + s_{i+1} \le \bar{v}_i + \bar{v}_{i+1} + s_{i+2} \le \cdots \le \bar{v}_i + \bar{v}_{i+1} + \bar{v}_{i+2} + \cdots \le n_0,$$

where the last inequality holds because the number of columns j with sum greater or equal to i is no more than the total number of nonempty columns. Since the addition of two d-digit numbers takes $\Theta(d)$ time, and there are $\binom{n_0+2}{2}$ binomial coefficients with entries less or equal to n_0, then the bound $\log_{10}\binom{j}{k} + 1 \le n_0 \log_{10} 2 + 1$ on the number of digits for such a binomial coefficient shows that this precomputation can be done in $O(n_0^3)$ time. Except in trivial cases (when the largest column sum is 1), the additional time needed does not affect the bounds on $T(\mathbf{p}, \mathbf{q})$ that we will prove below.

We now bound the time required for a given call to the algorithm.

LEMMA 6.3 (Time per call). $\tau(\mathbf{u}, \bar{\mathbf{v}}) = O((ab + c)(\log c)^3 |C_b(u_1)|)$ *for* $(\mathbf{u}, \bar{\mathbf{v}}) \in \mathcal{D}(\mathbf{p}, \mathbf{q})$, *where* $a = \max p_i$, $b = \max q_i$ *and* $c = \sum p_i$.

PROOF. See Miller and Harrison (2011). Due to space constraints, the proof has been omitted. □

LEMMA 6.4. $\#\Delta_k(j) \le \binom{j+k-1}{k-1}$ *for any* $j, k \in \mathbb{N}$.

PROOF. The map $f(\mathbf{r}) = (1r_1, 2r_2, \ldots, kr_k)$ is an injection $f : \Delta_k(j) \to C_k(j)$. Thus $\#\Delta_k(j) \le \#C_k(j) = \binom{j+k-1}{k-1}$. □

We are now ready to prove Theorem 3.3.

PROOF OF THEOREM 3.3 FOR $N(\mathbf{p}, \mathbf{q})$. By storing intermediate results in a lookup table, once we have computed $\bar{N}(\mathbf{u}, \bar{\mathbf{v}})$ upon our first visit to node $(\mathbf{u}, \bar{\mathbf{v}})$, we can simply reuse the result for later visits. Hence, we need only expend $\tau(\mathbf{u}, \bar{\mathbf{v}})$ time for each node $(\mathbf{u}, \bar{\mathbf{v}})$ occuring in the graph. Let $t_j = \sum_{i=j}^{m} p_i$ and $d = (ab + c)(\log c)^3$. Then

$$
\begin{aligned}
T(\mathbf{p}, \mathbf{q}) = \sum_{(\mathbf{u}, \bar{\mathbf{v}}) \in \mathcal{D}(\mathbf{p}, \mathbf{q})} \tau(\mathbf{u}, \bar{\mathbf{v}}) &\overset{(a)}{\le} \sum_{j=1}^{m} \sum_{\bar{\mathbf{v}} \in \Delta_b(t_j)} \tau\big(L^{j-1}\mathbf{p}, \bar{\mathbf{v}}\big) \\
&\overset{(b)}{\le} \sum_{j} \sum_{\bar{\mathbf{v}}} \mathrm{O}\big(d|C_b(p_j)|\big) = \sum_{j} \mathrm{O}\big(d|C_b(p_j)||\Delta_b(t_j)|\big) \\
&\overset{(c)}{\le} \sum_{j} \mathrm{O}\left(d \binom{p_j + b - 1}{b - 1} \binom{t_j + b - 1}{b - 1}\right) \\
&\overset{(d)}{\le} \sum_{j} \mathrm{O}\left(d \binom{a + b - 1}{b - 1} \binom{c + b - 1}{b - 1}\right) \\
&\le \mathrm{O}\big(dm(a + b - 1)^{b-1}(c + b - 1)^{b-1}\big),
\end{aligned}
$$

where (a) follows by Lemma 6.2, (b) by Lemma 6.3, (c) by Lemma 6.4 and (d) since $p_j \le a$ and $t_j \le c$. This proves (1) and (2). Now, (3) and (4) follow from (2) since $a \le c \le bn$. □

PROOF OF THEOREM 3.3 FOR $M(\mathbf{p}, \mathbf{q})$. Other than the coefficients, the only difference between the recursion for $\bar{M}(\mathbf{p}, \bar{\mathbf{q}})$ and that for $\bar{N}(\mathbf{p}, \bar{\mathbf{q}})$ is that we are summing over $\mathbf{s}$ such that $\mathbf{s} \in C^{\mathbf{r}+L\mathbf{s}}(p_1)$. Lemma 6.2 holds with the same proof, except with $C^{\mathbf{r}^1}(p_1), \ldots, C^{\mathbf{r}^{j-1}}(p_{j-1})$ replaced by $C^{\mathbf{r}^1+L\mathbf{s}^1}(p_1), \ldots, C^{\mathbf{r}^{j-1}+L\mathbf{s}^{j-1}}(p_{j-1})$, respectively. Lemma 6.3 also continues to hold; see Miller and Harrison (2011) for details. Consequently, the proof of the bounds goes through as well. □

This completes the proof of Theorem 3.3. Now we address the time required to uniformly sample a matrix with specified margins. Let $T_r(k)$ be the maximum over $1 \le j \le k$ of the expected time to generate a random integer uniformly

between 1 and j. If we are given a random bitstream [independent and identically distributed Bernoulli(1/2) random variables] with constant cost per bit, then $T_r(k) = \mathrm{O}(\log k)$, since for any $j \le k$, $\lceil \log_2 j \rceil \le \lceil \log_2 k \rceil$ random bits can be used to generate an integer uniformly between 1 and $2^{\lceil \log_2 j \rceil}$, and then rejection sampling can be used to generate uniform samples over $\{1, \ldots, j\}$. Since the expected value of a Geometric(p) random variable is $1/p$, then the expected number of samples required to obtain one that falls in $\{1, \ldots, j\}$ is always less than 2. More generally, for any fixed $d \in \mathbb{N}$, if we can draw uniform samples from $\{1, \ldots, d\}$, then we have $T_r(k) = \mathrm{O}(\log k)$ by considering the base-d analogue of the preceding argument.

LEMMA 6.5 (Sampling time). *Algorithm 3.4 takes*

$$\mathrm{O}\big(mT_r\big(n^c\big) + maT_r(n) + mb\log(a+b)\big)$$

expected time per sample in the binary case, and

$$\mathrm{O}\big(mT_r\big((2c)^c\big) + maT_r(n) + mb\log(a+b)\big)$$

expected time per sample in the $\mathbb{N}$-valued case. If $T_r(k) = \mathrm{O}(\log k)$, then this is $\mathrm{O}(mc\log c)$ expected time per sample in both cases.

REMARK. If b is bounded then $\mathrm{O}(mc\log c) \le \mathrm{O}(mn\log n)$ since $c \le bn$, and so this is polynomial expected time for bounded column sums.

PROOF. See Miller and Harrison (2011). Due to space constraints, the proof has been omitted. □

APPENDIX: SENSITIVITY ANALYSIS

Here, we examine the sensitivity of the ecology results in Section 4.1 to data-collection errors and to the choice of test statistic. First, it is quite possible that co-occurrence matrices such as the finch data in Table 4 may contain some data-collection errors; in particular, it is conceivable that a given species does in fact inhabit a particular island, but was not seen by the observers. We analyze the sensitivity of the p-value to such errors, and find that it is not particularly sensitive. Second, the particular choice of test statistic should not strongly influence the results. We assess the sensitivity of the p-value to variations of the test statistic.

A.1. Sensitivity to data-collection errors. Given $d \in \{1, 2, \ldots\}$, consider the set of binary matrices that can be obtained from Table 4 by flipping d zeros to ones. Each such matrix has a p-value under the test statistic $\overline{S^2}$. For $d \in \{1, 2, 3, 4, 5, 10\}$, and for a range of threshold values $\alpha \in [0, 0.05]$, we estimated the proportion of matrices with a p-value exceeding α; see Figure 3(a). Figure 3(b) contains (exact)

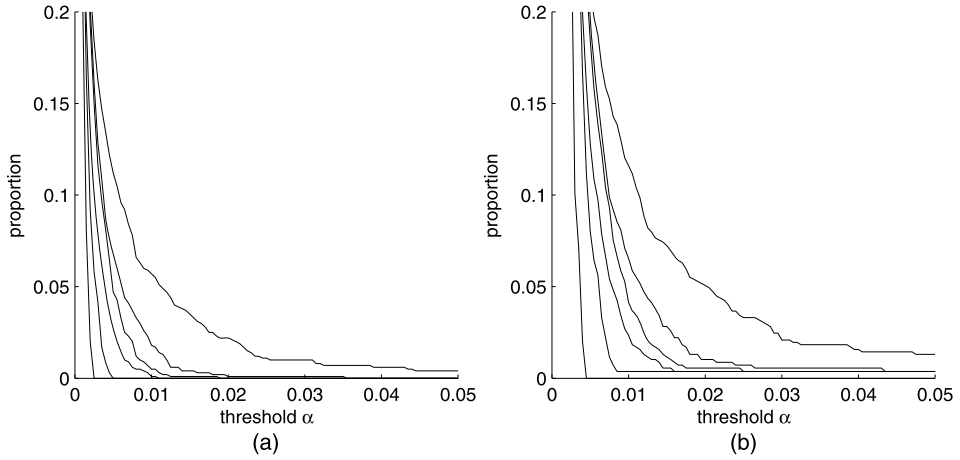

FIG. 3. (a) *Estimated proportion of p-values exceeding* α, *and* (b) 95% *confidence upper bound on this proportion. From bottom to top, the curves are* $d = 1, 2, 3, 4, 5, 10$.

95% confidence upper bounds on these estimates, for any given threshold α (not for all thresholds simultaneously). See below for details.

These results suggest that the proportions are very small—indeed, the estimated proportion exceeding $\alpha = 0.05$ is zero for all $d \leq 5$. This indicates that the p-value is not highly sensitive to data-collection errors of this kind. We conducted the same analysis over all binary matrices within Hamming distance d (although it seems unlikely that a species would be mistakenly recorded as present), and the estimates are somewhat larger—particularly for the case of $d = 10$—but remain small for $d \leq 5$.

For $d > 1$, the estimates were made by uniformly drawing $L = 1000$ matrices at distance d and, for the ith matrix, using $N = 10^4$ samples from our algorithm to compute an estimate $\hat{\theta}_i$ of the p-value. For each d, Figure 3(a) shows α versus $L^{-1}\sum_{i=1}^{L} I(\hat{\theta}_i > \alpha)$. Let $B_{N,\beta}(x) = \sup\{\theta \in [0,1] : F(x; N, \theta) \geq \beta\}$, where $F(x; N, \theta) = \mathbb{P}(X \leq x)$ with $X \sim \text{Binomial}(N, \theta)$. Figure 3(b) shows α versus $B_{L,\gamma}(\sum_{i=1}^{L} I(B_{N,\beta}(N\hat{\theta}_i) > \alpha))$, for $\gamma = 0.025$ and $\beta = 1 - (1-\gamma)^{1/L}$. The case $d = 1$ is slightly different because we exhaustively explore all 99 possible matrices instead of sampling L of them.

A.2. Sensitivity to test statistic variations. The idea behind the Roberts and Stone (1990) test statistic $\overline{S^2} = \binom{m}{2}^{-1}\sum_{i<j} s_{ij}^2$ is as follows. Uniformly choose a pair of species, and let X be the number of habitats shared [i.e., X is s_{ij} with probability $1/\binom{m}{2}$]. Roberts and Stone (1990) argue that, although exceptions can be constructed, competition/cooperation effects will typically make $\text{Var}(X)$ larger (relative to other co-occurrence matrices with the same margins), since they will tend to make the s_{ij} values more extreme. Using $\text{Var}(X)$ as a test statistic is equiv-

TABLE 8
Effect of the test statistic on the p-value for Table 4

c	Estimated p-value	95% confidence interval
0.5	0.162	[0.159, 0.164]
1	0.0113	[0.0106, 0.0120]
2	0.00044	[0.00032, 0.00059]
3	0.00016	[0.00009, 0.00026]

alent to using $\overline{S^2}$, since $\mathbb{E}(X^2) = \overline{S^2}$ and $\mathbb{E}(X)$ is the same for all matrices with the same margins.

It seems that the same argument could be used to justify any statistic of the form $\mathbb{E}f(|X - \mathbb{E}X|)$, where f is monotone increasing on $[0, \infty)$. Of course, $\mathrm{Var}(X)$ is the case of $f(x) = x^2$. To study sensitivity to the choice of f, we estimated the p-value of the finch data (Table 4) with $f(x) = x^c$ for $c \in \{0.5, 1, 2, 3\}$, using 10^5 samples. See Table 8.

Interestingly, the p-value with $c = 0.5$ is quite large. This suggests that, in this case at least, the more extreme values of s_{ij} are playing an important role. Whether or not this is scientifically relevant is a question for ecologists.

Acknowledgement. We would like to thank the Associate Editor for suggesting the sensitivity analysis in the Appendix.

REFERENCES

ANAND, H., DUMIR, V. C. and GUPTA, H. (1966). A combinatorial distribution problem. *Duke Math. J.* **33** 757–769. MR0201329

ATMAR, W. and PATTERSON, B. D. (1995). The nestedness temperature calculator: A visual basic program, including 294 presence–absence matrices. AICS Research Incorporated and The Field Museum. Available at http://www.aics-research.com/nestedness/tempcalc.html.

BARVINOK, A. I. (1994). A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.* **19** 769–779. MR1304623

BECK, M. and PIXTON, D. (2003). The Ehrhart polynomial of the Birkhoff polytope. *Discrete Comput. Geom.* **30** 623–637. MR2013976

CANFIELD, E. R., GREENHILL, C. and MCKAY, B. D. (2008). Asymptotic enumeration of dense 0–1 matrices with specified line sums. *J. Combin. Theory Ser. A* **115** 32–66. MR2378856

CANFIELD, E. R. and MCKAY, B. D. (2005). Asymptotic enumeration of dense 0–1 matrices with equal row sums and equal column sums. *Electron. J. Combin.* **12** Research Paper 29, 31 pp. (electronic). MR2156683

CHEN, Y., DIACONIS, P., HOLMES, S. P. and LIU, J. S. (2005). Sequential Monte Carlo methods for statistical analysis of tables. *J. Amer. Statist. Assoc.* **100** 109–120. MR2156822

CONNOR, E. F. and SIMBERLOFF, D. (1979). The assembly of species communities: Chance or competition? *Ecology* **60** 1132–1140.

DE LOERA, J. A. and STURMFELS, B. (2003). Algebraic unimodular counting. *Math. Program.* **96** 183–203. MR1993046

DE LOERA, J. A., HEMMECKE, R., TAUZER, J. and YOSHIDA, R. (2004). Effective lattice point counting in rational convex polytopes. *J. Symbolic Comput.* **38** 1273–1302. MR2094541

DIACONIS, P. and EFRON, B. (1985). Testing for independence in a two-way table: New interpretations of the chi-square statistic. *Ann. Statist.* **13** 845–913. MR0803747

DIACONIS, P. and GANGOLLI, A. (1995). Rectangular arrays with fixed margins. In *Discrete Probability and Algorithms* (*Minneapolis*, *MN*, 1993). *IMA Vol. Math. Appl.* **72** 15–41. Springer, New York. MR1380519

DIAMOND, J. M. (1975). Assembly of species communities. In *Ecology and Evolution of Communities* 342–444. Harvard Univ. Press, Cambridge, MA.

DYER, M., KANNAN, R. and MOUNT, J. (1997). Sampling contingency tables. *Random Structures Algorithms* **10** 487–506. MR1608222

GAIL, M. and MANTEL, N. (1977). Counting the number of $r \times c$ contingency tables with fixed margins. *J. Amer. Statist. Assoc.* **72** 859–862. MR0461792

GALE, D. (1957). A theorem on flows in networks. *Pacific J. Math.* **7** 1073–1082. MR0091855

GALTON, F. (1889). *Natural Inheritance*. MacMillan, New York.

GESSEL, I. M. (1987). Enumerative applications of symmetric functions. *Séminaire Lotharingien de Combinatoire* **B17a** 5–21.

GESSEL, I. M. (1990). Symmetric functions and P-recursiveness. *J. Combin. Theory Ser. A* **53** 257–285. MR1041448

GOTELLI, N. J. and MCCABE, D. J. (2002). Species co-occurrence: A meta-analysis of JM Diamond's assembly rules model. *Ecology* **83** 2091–2096.

GREENHILL, C., MCKAY, B. D. and WANG, X. (2006). Asymptotic enumeration of sparse 0–1 matrices with irregular row and column sums. *J. Combin. Theory Ser. A* **113** 291–324. MR2199276

HARRISON, M. T. and MILLER, J. W. (2013). Importance sampling for weighted binary random matrices with specified margins. Preprint. Available at arXiv:1301.3928 [stat.CO].

HOLMES, R. B. and JONES, L. K. (1996). On uniform generation of two-way tables with fixed margins and the conditional volume test of Diaconis and Efron. *Ann. Statist.* **24** 64–68. MR1389880

JOHNSEN, B. and STRAUME, E. (1987). Counting binary matrices with given row and column sums. *Math. Comp.* **48** 737–750. MR0878703

MACMAHON, P. A. (1915). *Combinatory Analysis I*, *II*. Cambridge Univ. Press, London.

MCKAY, B. D. (1983). Applications of a technique for labelled enumeration. In *Proceedings of the Fourteenth Southeastern Conference on Combinatorics*, *Graph Theory and Computing* (*Boca Raton*, *Fla.*, 1983) *Congr. Numer.* **40** 207–221. MR0734371

MCKAY, B. D. and WORMALD, N. C. (1990). Uniform generation of random regular graphs of moderate degree. *J. Algorithms* **11** 52–67. MR1041166

MILLER, J. W. and HARRISON, M. T. (2011). Exact enumeration and sampling of matrices with specified margins. Unpublished manuscript. Available at arXiv:1104.0323 [stat.CO].

MOUNT, J. (2000). Fast unimodular counting. *Combin. Probab. Comput.* **9** 277–285. MR1774755

PATTERSON, B. and ATMAR, W. (1986). Nested subsets and the structure of insular mammalian faunas and archipelagos. *Biological Journal of the Linnean Society* **28** 65–82.

PÉREZ-SALVADOR, B. R., DE-LOS COBOS-SILVA, S., GUTIÉRREZ-ANDRADE, M. A. and TORRES-CHAZARO, A. (2002). A reduced formula for the precise number of $(0,1)$-matrices in $\mathscr{A}(\mathbf{R},\mathbf{S})$. *Discrete Math.* **256** 361–372. MR1927558

READ, R. C. (1959). The enumeration of locally restricted graphs. I. *J. London Math. Soc.* **34** 417–436. MR0108446

READ, R. C. (1960). The enumeration of locally restricted graphs. II. *J. London Math. Soc.* **35** 344–351. MR0140443

REDFIELD, J. H. (1927). The theory of group-reduced distributions. *Amer. J. Math.* **49** 433–455. MR1506633

ROBERTS, A. and STONE, L. (1990). Island-sharing by archipelago species. *Oecologia* **83** 560–567.

RYSER, H. J. (1957). Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.* **9** 371–377. MR0087622

STANLEY, R. P. (1973). Linear homogeneous Diophantine equations and magic labelings of graphs. *Duke Math. J.* **40** 607–632. MR0317970

ULRICH, W. and GOTELLI, N. J. (2007). Null model analysis of species nestedness patterns. *Ecology* **88** 1824–1831.

WANG, B. Y. (1988). Precise number of (0, 1)-matrices in $U(R, S)$. *Scientia Sinica*, *Series A* **XXXI** 1–6.

WANG, B.-Y. and ZHANG, F. (1998). On the precise number of (0, 1)-matrices in $U(R, S)$. *Discrete Math.* **187** 211–220. MR1630720

DIVISION OF APPLIED MATHEMATICS
BROWN UNIVERSITY
PROVIDENCE, RHODE ISLAND 02912
USA
E-MAIL: Jeffrey_Miller@Brown.edu
Matthew_Harrison@Brown.edu