Short title: SAMPLING FEASIBLE SETS OF ECOLOGICAL PATTERNS

Article title: Efficient algorithms for sampling feasible sets of abundance distributions

Kenneth J. Locey[‡1] and Daniel J. McGlinn[*1,2]

Department of Biology, Utah State University, Logan, UT, 84322

*Ecology Center, Utah State University, Logan, UT, 84322

‡To whom correspondence should be addressed

1 ken@weecology.org; phone: (435) 764-5070, fax (435) 797-1575

2 daniel.mcglinn@usu.edu;

Statement of authorship: KL derived the algorithms, coded them in Python, conducted the analyses, and served as primary author. DM coded algorithms into R and provided an R package distributed on the comprehensive R archive network (CRAN) and served as secondary author. Both KL and DM contributed to the organization of the paper and its ideas.

Keywords: abundance patterns, constraints, feasible set, integer partition, macroecology, sampling algorithms, species abundance distribution

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

SUMMARY

Ecological variables such as species richness (*S*) and total abundance (*N*) can strongly influence ecological patterns. For example, the general form of the species abundance distribution (SAD) can often be explained by the majority of possible forms having the same *N* and *S*, i.e. the SAD feasible set. The feasible set reveals how variables determine observable variation, whether empirical patterns are exceptional to the majority of possible forms, and provides a constraint-based explanation for the ubiquity of hollow-curve SADs in nature. However, use of the feasible set has been limited to inefficient sampling algorithms that prevent large ecological communities and ecologically realistic combinations of *N* and *S* from being examined. This is the primary hindrance to using this otherwise novel perspective and theoretical framework. We developed efficient computational algorithms to generate random samples of the feasible set for the SAD and similar discrete distributions of abundance, including those that allow for zero-values, e.g., absences. We provide Python and R based implementations of our algorithms and tools for testing and using them. Our algorithms are often several orders of magnitude faster than a long-standing and recently used approach. This greatly increases the size and diversity of communities that can be examined with the feasible set approach and thus advances progress using constraint-based approaches to decipher ecological patterns.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

The species abundance distribution (SAD) is an important ecological pattern that characterizes the interspecific pattern of commonness and rarity across a community (Brown 1995; Rosenzweig 2002; Blackburn & Gaston 2003). The SAD has been used to gain insight into the underlying mechanisms shaping communities, and the ability to predict the shape of the SAD has served as a primary basis for comparing biodiversity theories (e.g. MacArthur & Wilson 1967; Scudo & Ziegler 1978; May 1981; Hubbell 2001; Harte 2011). Though theoretical explanations of the SAD are underpinned by a diverse array of processes such as colonization and dispersal limitation, niche differentiation, metacommunity dynamics, and stochastic population dynamics (e.g. Hubbell 2001, Leibold et al. 2004, McGill 2010), it has become increasingly clear that general variables such as total community abundance (N) and species richness (S) have strong constraining influences on the form of the SAD, regardless of the mechanisms operating in the community (Harte *et al*. 2008; McGlinn & Hurlbert 2012; Supp *et al*. 2012; White *et al*. 2012, Locey & White 2013). For example, Locey and White (2013) demonstrated that most of the possible shapes of the SAD having the same N and S, i.e., the SAD feasible set, capture the majority of variation in empirical SADs.

The SAD feasible set reveals how N and S numerically constrain observable variation in the form of the SAD and whether empirical SADs are exceptional to or representative of the majority of possible SAD shapes (e.g. rank-abundance curves, frequency distributions) for a given set of constraint variables (e.g. N, S). Likewise, the general properties of the feasible set reveal simple explanations for common patterns. For example, the vast majority of possible shapes of the SAD for ecologically realistic values of the N and S are hollow-curves, i.e., frequency distribution revealing a trend of decreasing frequency with increasing abundance classes. This provides a simple and straightforward first-principle explanation for the ubiquity of

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

hollow-curve SADs in nature. Though only the SAD has been examined using this type of feasible set approach (i.e. all possible forms of a pattern), Locey & White (2013) suggest that other ecological patterns could also be examined in the context of their feasible sets. However, there are computational challenges to using their approach.

Feasible sets can be immense and enumerating them can be untenable. However, small random samples can be used to characterize the center of the feasible set (i.e., average form) as well as the distribution of statistical features (e.g., evenness, diversity) within it. Locey & White (2013) took a conceptually simple approach to sampling the SAD feasible set, an approach known as integer partitioning. This approach is based on the fact that there are a limited number of unordered ways that the abundances of S unlabeled species can sum to a total abundance of N unlabeled individuals. These unordered configurations of integers are called integer partitions (Bóna 2006). For example, the feasible set of integer partitions for $N = 6$ and $S = 3$ is: {(4, 1, 1), (3, 2, 1), (2, 2, 2)}, where differently ordered configurations having the same integer values (e.g. (4, 1, 1), (1, 1, 4), (1, 4, 1)) represent the same integer partition (4, 1, 1). Important to note, is that each integer partition represents a unique rank-abundance curve, e.g., (4, 1, 1) that corresponds one-to-one with a unique frequency distribution, e.g., (one 4 and two 1's).

Use of integer partitioning to randomly sample the feasible set allows the feasible space to be characterized without generating all possible forms. However, all published partitioning algorithms sample the feasible set only with regards to the total, N. In this way, all partitions of N have the same probability of being drawn, regardless of the number of elements S. This means that randomly sampling the feasible set for a given N and S, requires generating partitions according to N and then rejecting those not having S elements, often resulting in impractically high rejection rates. For example, randomly generating one partition for $N = 1000$ and $S = 10$

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

requires drawing from a feasible set of nearly 2.4×10^{31} partitions, one of the roughly 8.9×10^{14} having 10 elements (a probability of nearly 3.7×10^{-17}); a practically impossible task.

Another challenge in applying integer partitioning to the study of feasible sets is that some ecological patterns of abundance include parts with zero values. One example is the species spatial abundance distribution (SSAD) describing the frequency with which individuals of a single species occupy areas within a landscape (Brown *et al*. 1995; Harte *et al*. 2008; Haegeman & Etienne 2010; Harte 2011). The SSAD reflects the spatially implicit pattern of aggregation of individuals across a landscape and is mechanistically linked to other ecological patterns (Brown *et al*. 1995; Harte 2011). In the SSAD, individuals can be absent from a number of areas, meaning that there are some areas with zero individuals. Because integer partitions *per se* do not include zeros, integer partitioning methods need to be modified to examine the SSAD feasible set and potentially other patterns that include zeros.

Here, we present algorithms that greatly increase the efficiency of sampling the SAD feasible set as well as feasible sets that include zeros. We explain each algorithm in concept and develop Python and R based packages to implement them. We test each algorithm for sampling bias and for speed against the method of Locey & White (2013). To reveal the practical gains of these new algorithms, we reanalyze the SAD datasets of Locey & White (2013), one of the largest and most diverse compilations of species abundance data, wherein it took more than 10000 compute hours to examine only 60% of the available data (9562 of 15950 SADs). Our new algorithms allow us to examine a larger portion of the data in less than one tenth the time. Our work expands the feasible set approach to values of N and S that were previously untenable and to new patterns of abundance, thus advancing the front of a numerical constraint-based

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

approach to understanding and quantifying ecological patterns (Pueyo *et al*. 2007, Haegeman & Loreau 2008, Harte *et al*. 2008, Haegeman & Etienne 2010, Harte 2011, Locey & White 2013).

METHODS

We develop integer partitioning algorithms that generate random samples of feasible sets for discrete distributions of abundance that are defined by a total N and composed of S elements, where both N and S are positive integers. Integer partitioning is a mature field of mathematics and algorithms for generating random partitions of N (e.g. RANPAR of Nijenhuis & Wilf 1978) are often implemented in mathematical environments (e.g. Sage, Maple, Mathematica). However, these algorithms and software do not randomly partition N into exactly S elements. Instead, they partition N into a random number of elements ranging from 1 to N, and then maybe, do this many times until a match to S is found. This requires unreasonably high rejection rates for ecologically realistic combinations of N and S. Here, we develop an approach to generate random partitions of N having exactly S elements. We begin by noting a basic integer partitioning relationship:

1.) For every integer N there are $q(N, S)$ partitions having S or less elements as well as $q(N, S)$ partitions having S or less as the first element (Bóna 2006).

Computationally the function $q(N, S)$ is given by the recurrence relation:

$$q(n,k) = q(n,k-1) + q(n-k,k)$$

where the number of partitions of n having k or less elements equals the sum of the number of partitions of n having k - 1 or less elements and the number of partitions n - k having k or less elements and where, by convention, $q(n, 0) = 0$, $q(1, k) = 1$.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

To better understand this relationship consider the five partitions of the integer 4: {(4), (3, 1), (2, 2), (2, 1, 1), (1, 1, 1, 1)}. Each partition can be represented by rows of dots called Ferrers diagrams, where each row represents an element in the partition (Fig 1). The Ferrers diagrams (Fig1) demonstrate that the number of partitions of $N = 4$ with, say, $S = 3$ or less as elements equals the number of partitions of 4 having 3 or less elements. This raises another important relationship: each partition of N with S elements corresponds to exactly one partition of N with S as the largest element (Bóna 2006). Rotating a partition about its upper left to lower right diagonal reveals this (Fig 1). In some cases, rotating a partition about its diagonal produces the same partition, e.g. (2, 2) is a self-conjugate. This is a classic proof by bijection (one-to-one correspondence) for the above relationship. As a result, the value of the first element in a partition always determines the number of elements in its conjugate (Bóna 2006). We will use this information to generate random partitions of N having exactly S elements.

Here, we describe how to build a random partition of N having S as the largest element element-by-element, and finally conjugate the resulting partition to produce a random partition of N having S elements. This is a far simpler problem than directly generating a random partition of N having S elements, which seems intractable by the lack of solutions for it in combinatorial texts (including Bóna 2006), peer-reviewed research, preprints on Arxiv.org, and the lack of any functions for it in well-known mathematical software such as Sage, Matlab, Maple, and Mathematica. Now, because we know that we are only interested in partitions of N having S as the largest element, we can simply let the first element of our random partition be S. The conjugate will still be any one of the possible partitions of N having S elements. Having determined what the first part of our partition is, we then subtract S from N because the value of S is now part of the partition we are building. From here, we can generate a random partition

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

for the reduced value of N, where the value of the first element is no greater than S. We can do this, element-by-element, by iteratively applying relationship (1).

Here, we will find the first element of a randomly chosen partition of N having S or less as the largest element. We begin by choosing a whole number x at random ranging from 1 to $q(N, S)$ to represent any one of the $q(N, S)$ partitions of N having S or less as the first element. We then find the value of first element of the partition by 'sandwiching' it. First, we know there are at least x partitions of N having S or less as the first element, i.e., $x \leq q(N, S)$. Second, unless $x = 1$ (whereby the partition is just a vector of 1's), x must be greater than some candidate value C for which there are $q(N, C)$ partitions of N having C or less as the first element. Putting these two statements together, there must be a candidate value C that *violates* the inequality: $q(N, C) < x \leq q(N, S)$. In this way, we can find the value of the next element by trying values of C. Once we find the value, we append it to our partition and then decrease x by $q(N, C-1)$ and N by C. Iterating, we then find the value of the first element for our new combination of $(x - q(N, C-1))$ and $(N - C)$ by exploring candidate values no greater than C. Repeating this process sequentially builds the partition until $N = 0$ at which point the sum of the partition will equal the original value of N (Fig 1). Finally, we conjugate the partition and because it already has S as the first element, we produce a random partition of N (i.e. all are equally likely) having exactly S elements.

The question remains as to which candidate value to start with and how to proceed to different values. We could start with the smallest possible value of the largest part and take a 'bottom-up' approach, or the largest possible value and take a 'top-down' approach, or even choose a candidate value at random and use a 'divide-and-conquer' method. However, in the event that S is relatively small compared to N, e.g., by two or three orders of magnitude, these

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

approaches would be inefficient because they would first generate a partition of, say, $N = 200000$ having $S = 300$ as the first element, but having as many as 199701 elements! Clearly building the partition one element at a time would be inefficient in this case.

Here, we provide an alternative and, likewise, novel algorithm to building the partition to obtain a random partition of N having S elements. This is not the primary method of sampling a feasible set because it is not always the most efficient approach due to the computational overhead. This method builds a partition of N having exactly S elements using multiples of integers – the 'multiplicity approach'. Instead of finding the value of the first element, appending it, and moving on to find the next value, we can instead ask how many times does the particular value occur, e.g. 1 occurs twice in the partition (2, 1, 1). We can start with the smallest possible multiple (i.e. $m = 1$) and ask whether x is less than or equal to the number of partitions of $N - S*m$ having less than S as the first part. This is because the set of partitions of N having a number of S's equal to m actually contains the set of partitions of $N - S*m$ having less than S as the first part (see Appendix 1). We can then increase m by one until $x \leq q(N - S*m, S)$, at which point we will have found the corresponding multiple of k. Note again, values of N, S, and x are reduced as the partition is built, lest we never stop building it.

Random partitions with some elements having zero values

The above algorithms address distributions having positive values, such as the distribution of abundance among species (SAD). In contrast, some ecological patterns include zero values (e.g. absences). One example is the species spatial abundance distribution (SSAD), a frequency distribution that characterizes the number of quadrats, cells, or areas containing a given abundance of a species (Brown *et al*. 1995; Haegeman & Etienne 2010; Harte 2011). However,

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

only small changes are needed to adapt the above approaches to cases allowing zero-valued parts. For example, let 10 unlabeled individuals occupy a landscape sectioned into quarters. The most aggregated distribution would be for all 10 to occupy the same quarter, [10, 0, 0, 0]. The least aggregated would be for 3 to occupy two quarters while 2 occupy the other two quarters, [3, 3, 2, 2]. In fact, the number of configurations for 10 unlabeled individuals distributed across 4 unlabeled sections equals the number of partitions of 10 having 4 or less parts, i.e. $q(10, 4) = 23$. Consequently, if the number of parts S is less than the total N, then a random partition would simply be a random partition for N having S or less parts, with zeros appended to ensure the final form of the partition has S parts.

On the other hand if the number of parts S is greater than the total N, then a different approach is needed. To see this let 4 unlabeled individuals occupy a landscape sectioned into tenths. The most aggregated distribution would be for all 4 to occupy the same subsection, [4, 0, 0, 0, 0, 0, 0, 0, 0, 0] and the least aggregated configuration would be for 4 sections to have one individual and for 6 sections to have zero, i.e. [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]. In this way, the number of possible configurations for 4 unlabeled individuals distributed across 10 unlabeled sections is simply $q(N, N)$, once again, because we are dealing with unordered configurations of unlabeled elements. Consequently, if $N < S$, a random partition for N and S, allowing for zero-valued parts, is simply a random partition for N having N or less parts, with zeros appended to ensure the partition has S parts.

Examining for bias and speed

We implemented the above algorithms in Python and R and made them freely available using a public GitHub repository (https://github.com/klocey/partitions). We also developed these

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

algorithms into an R package, rparitions. To further ensure that our sampling algorithms were unbiased we used kernel density curves to visually compare the results of the above algorithms to full feasible sets and random samples generated with the function implemented in the Sage mathematical environment (www.sagemath.org) that is based on the RANPAR algorithm of Nijenhuis & Wilf (1978) (note: where the number of elements S is an output) and is the method used in Locey & White (2013). Our Python package include test files that conduct, among several other tests of our source code and partitioning functions, 2-sample t-tests and 2-sample Kolmogorov Smirnov tests on kernel density curves of the variance of logarithmically transformed abundances from random samples generated by the Sage software and random samples generated using the algorithms derived here. These tests are important to regularly run because additional code developments can corrupt source code.

Though we provide a recoding of the RANPAR algorithm in our software, to free users from having to use Sage, our comparisons are based on using the Sage software to draw random samples using the RANPAR algorithm. This is largely because Sage is an impressively powerful software and we were, in part, interested in the practical gains of our algorithms over it. If our algorithms are unbiased, then their distributions will not differ in any systematic way from full feasible sets and random samples generated using the proven function implemented in Sage.

We compare the computational speed of our algorithms to that of the approach used in Locey & White (2013) (i.e. using Sage to generate random partitions for a given q and rejecting those not having n elements) across a range of values of q, n, and q-n ratios for which the latter method was likely to return random samples within reasonable time (one hour). Because Sage is coded in Python, our comparisons are made using the Python versions of our algorithms.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

Empirical Demonstration of the New Algorithms

Locey and White (2013) analyzed the species abundance distributions (SADs) of 9562 sites of trees, bird, mammal, fungi, and prokaryote communities using a partitioning algorithm that sampled the feasible set according to total abundance N but not with respect to species richness S (i.e. the number of elements). Those data consisted, in part, of a subset of previously compiled datasets of site-specific species abundance data (see White *et al*. 2012), and included four continental-to-global scale surveys, including the Christmas Bird Count (129 sites) (National Audubon Society 2002), North American Breeding Bird Survey (1,586 sites) (Sauer et al. 2011), Gentry's Forest Transect Data Set (182 sites) (Phillips and Miller 2002), Forest Inventory Analysis (7,359 sites) (U.S. Department of Agriculture 2010), and one global-scale data compilation, the Mammal Community Database (42 sites) (Thibault et al. 2011). Locey and White (2013) also compiled abundance data at the species level from five microbial metagenome projects for a total of 264 SADs. Those data were obtained from the metagenomics server MG-RAST (Meyer et al. 2008). Metagenomic data were compiled into datasets representing aquatic prokaryotic communities (48 metagenomes) (Flores et al. 2011, www.catlin.com/en/Responsibility/CatlinArcticSurvey), terrestrial prokaryotic communities (92 metagenomes) (Chu et al. 2010, Fierer et al. 2012), and terrestrial fungal communities (124 metagenomes) (Amend et al. 2010). We refer the reader to Locey & White (2013) and White *et al*. (2012) for more thorough descriptions of those datasets.

The inefficiency of the partitioning method used in Locey & White (2013) restricted their analyses to combinations of N and species richness S, for which, there was a reasonable probability of generating a random integer partition of N with exactly S elements. This restriction allowed for only 60% of the available data to be examined despite more than 10000 compute

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

hours worth of effort. We reanalyze those datasets using the algorithms developed here, which should allow for random samples of a greater number of SADs to be produced in less time.

RESULTS

Statistical properties of entire feasible sets are indistinguishable from random samples generated with our sampling algorithms, demonstrating that the implementations of our algorithms were unbiased (Fig 2 and Figs 1-2 of Appendix). When generating 300 random partitions, i.e. enough to safely characterize the feasible space (Locey & White 2013), these implementations were between 10 and 10^5 times faster than the method used by Locey & White (2013) for the combinations of N and S we tested (Fig 3). These combinations were limited to values of N and S for which the algorithm used in Sage could generate random samples in reasonable time. Each algorithm was best suited for particular values of N and S (Fig 4). For cases where all parts have positive values, the multiplicity algorithm is the fastest for combinations where N is partitioned among a relatively small number of elements (Fig 3 Appendix).

The greater efficiency of the algorithms developed here allowed us to generate between 300 and 500 random partitions for 92.7% of the SADs (14786/15950) from the compilation of SAD data used by Locey & White (2013), in less than 1000 compute hours. In contrast, the method used by Locey & White (2013) required more than 10000 compute hours to generate between 300 and 500 random partitions for 60% of the available data (9562/15950 SADs).

DISCUSSION

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

The feasible set approach provides a framework for understanding how constraints determine observable variation in ecological patterns and distributions of wealth and abundance. We greatly improved the efficiency of an integer partitioning approach for randomly sampling the feasible set. The algorithms we derived greatly increase the practical use of feasible set by decreasing computing time and by allowing feasible sets of distributions characterized by zero values, e.g., the species spatial abundance distribution, to be defined, explored, and used to understand variation in empirical patterns. We provided the algorithms in two computing languages used by ecologists, i.e., R and Python, and took steps to ensure the implementations of our algorithms are unbiased. We provided R and Python test scripts to detect computational errors that can result when a user modifies the code.

Integer partitioning is only one way to examine and randomly sample the feasible set of possible SAD and SSAD shapes, i.e., for a given set of constraints. Other possibilities include constraint-based programming (see http://cran.r-project.org/web/views/Optimization.html) and iterative random walks, such as that used by Haegeman & Loreau (2008). Those approaches may not require combinatorial problems to be solved and so may not suffer from the problem of combinatorial explosion (large increases in the size of the feasible set for small changes in the total q and number of elements). However, one benefit to the integer partitioning approach is that the random sampling algorithms are inherently unbiased and do not require 'burn-in' periods to produce independent samples. Integer partitioning also reveals properties such as the size of the feasible set, the distribution of statistical characteristics (e.g. species evenness, diversity, modal abundance class) across the feasible set, and a fundamental way to unify different ecological patterns such as the SAD and SSAD, i.e., by their similar mathematical structure. However, we

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

suggest that approaches such as those in Haegeman & Loreau (2008) should also be developed and compared to integer partitioning.

The feasible set approaches taken here, in Haegeman & Loreau (2008), and in Locey & White (2013) ignore biological and statistical mechanism and focuses entirely on observable variation in the shape of empirical patterns. Consistency of empirical patterns with the center of the feasible set suggests that the shapes of those patterns contain little information beyond that encoded by the constraints used to characterize the feasible set (Haegeman & Loreau 2008; Locey & White 2013). However, consistency with the feasible set does not mean that biological processes are not operating but rather that they may indirectly influence empirical patterns through their effects on constraints (Supp *et al*. 2012; White *et al*. 2012). Indeed, if the majority of variation in an ecological pattern can be explained by a few general variables, then understanding the forces, processes, or mechanisms driving the values of those variables would seem to be of primary importance (McGill 2010). Alternatively if empirical patterns occupy an uncommon area of the feasible set, e.g., in being exceptionally uneven, then biological processes or additional constraints may be relevant.

Our work greatly advances the ability of ecologists to characterize and explore observable variation in ecological patterns of abundance by greatly decreasing computational time. These advances allow combinations of constraint values to be examined that were previously out of reach, and hence, will help provide a greater understanding of the degree to which small combinations of general variables can explain the forms of ecological distributions (e.g. SAD, SSAD) and common ecological patterns (e.g. hollow-curve frequency distributions, Taylor's Law). The algorithms we developed apply to frequency distributions such as the SAD and SSAD. However, many ecological patterns are also cumulative, describing the rates at which

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

species are encountered with increasing area (species-area relationship) or time (species-time relationship) or both (species-time-area relationship), as well as the spatially implicit distribution of occupancy among species within a landscape (occupancy-frequency distribution). Characterizing and randomly sampling the feasible sets of these and other patterns may require modification of the algorithms we developed, approaches more similar to that of Haegeman and Loreau (2008), or altogether new approaches.

ACKNOWLEDGMENTS

We thank X. Xiao and E. P. White for critical discussions and friendly reviews. We thank the numerous individuals involved in collecting and providing the data used in this paper including the essential citizen scientists who collect the North American Breeding Bird Survey and Christmas Bird Count data, USGS and CWS scientists and managers, researchers who collected and sequenced the microbial metagenomic data, the MG-RAST project, the Ribosome Database Project, the Audubon Society, the U.S. Forest Service, the Missouri Botanical Garden, and Alwyn H. Gentry. DJM was supported by a CAREER grant from the U.S. National Science 303 Foundation to E.P. White (DEB-0953694).

LITERATURE CITED

Amend, A. S., Seifert, K. A., Samson, R., & Bruns, T.D. (2010). Indoor fungal composition is geographically patterned and more diverse in temperate zones than in the tropics. *P. Natl. Acad. Sci. USA*., 107, 13748-13753.

Bóna, M. 2006. A walk through combinatorics: An introduction to enumeration and graph theory. 2nd Edition. World Scientific Publishing Co. Singapore.

Brown, J. H. 1995. Macroecology. Univ. Chicago Press, Chicago.

Brown, J. H., Mehlman, D. W. and G. C. Stevens. 1995. Spatial variation in abundance. Ecology, 76, 2028-2043.

Chu, H., Fierer, N., Lauber, C.L., Caporaso, J.G., Knight, R. & Grogan, P. (2010). Soil bacterial diversity in the Arctic is not fundamentally different from that found in other biomes. *Environ*. *Microbiol*., 12,2998–3006.

Fierer, N., Lauber, C.L., Ramirez, K.S., Zaneveld, J., Bradford, M.A. & Knight, R. (2012). Comparative metagenomic, phylogenetic and physiological analyses of soil microbial communities across nitrogen gradients. *ISME J*., 6, 1007–17.

Flores, G.E., Campbell, J., Kirshtein, J., Meneghin, J., Podar, M., Steinberg, J.I. *et al*. (2011). Microbial community structure of hydrothermal deposits from geochemically different vent fields along the Mid-Atlantic Ridge. *Environ*. *Microbiol*., 13, 2158-2171.

Haegeman, B. and R. S. Etienne. 2010. Entropy Maximization and the Spatial Distribution of Species. Am. Nat., 175, E74–E90.

Haegeman, B. and M. Loreau. 2008. Limitations of entropy maximization in ecology. Oikos, 117, 1700–1710.

Haegeman, B., & Loreau, M. (2009). Trivial and non-trivial applications of entropy maximization in ecology: a reply to Shipley. *Oikos*, *118*(8), 1270-1278.

Harte, J., T. Zillio, E. Conlisk and A. B. Smith. 2008. Maximum entropy and the state-variable approach to macroecology. Ecology, 89, 2700–2711.

Locey, K. J. and E. P. White. 2013. How species richness and total abundance constrain the distribution of abundance. Ecology Letters, DOI: 10.1111/ele.12154.

MacArthur, R. H. and Wilson, E.O. (1967). *The theory of island biogeography* (Vol. 1). Princeton University Press.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

May, R. M. (1981). Theoretical ecology. Principles and applications. *Theoretical ecology. Principles and applications.*, (Ed. 2).

McGill, B.J. 2010. Towards a unification of unified theories of biodiversity. *Ecol. Lett.*, 13, 627–642.

McGill, B.J., Etienne, R.S., Gray, J.S., Alonso, D., Anderson, M.J., Benecha, H.K. *et al*. (2007). Species abundance distributions: moving beyond single prediction theories to integration within an ecological framework. *Ecol. Lett.*, 10, 995–1015.

McGlinn, D. J., and A. H. Hurlbert. 2012. Scale dependence in species turnover reflects variance in species occupancy. Ecology, 93, 294–302.

Meyer, F., Paarmann, D., D'Souza, M., Olson, R., Glass, E.M., Kubal, M. *et al*. (2008). The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9, 386.

National Audubon Society. (2002). The Christmas Bird Count historical results. Retrieved from http://www.audubon.org/bird/cbc.

Nijenhuis, A. and H. S. Wilf. 1978. Combinatorial Algorithms for Computers and Calculators. Academic Press, New York.

Pueyo, S., He, F. & Zillio, T. (2007). The maximum entropy formalism and the idiosyncratic theory of biodiversity. *Ecol. Lett.*, 10, 1017-1028.

Sauer, J.R., Hines, J.E., Fallon, J.E., Parkieck, D.J., Ziolkowski, D.J. Jr. & Link, W.A. (2011). *The North American Breeding Bird Survey 1966-2009*. Version 3.23.2011. USGS Patuxent Wildlife Research Center, Laurel, MD.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

Scudo, F. M., & Ziegler, J. R. (1978). *The Golden age of theoretical ecology, 1923-1940: a collection of works by V. Volterra, VA Kostitzin, AJ Lotka, and AN Kolmogoroff*. Springer-Verlag.

Stojmenovic, I. 2008. Generating all and random instances of a combinatorial object. In Handbook of Applied Algorithms: Solving scientific, Engineering and Practical Problems. John Wiley & Sons, Inc., New Jersey, pp. 1-38.

Storch, D., A. L. Šizling, J. Reif, J. Polechová, E. Šizlingová, and K. J. Gaston. 2008. The quest for a null model for macroecological patterns: geometry of species distributions at multiple spatial scales. Ecology Letters, 11,771–784.

Supp, S. R., X. Xiao, S. K. M. Ernest and E. P. White. 2012. An experimental test of the response of macroecological patterns to altered species interactions. Ecology, 93, 2505–2511.

Taylor LR (1961) Aggregation, variance and the mean. *Nature* 189, 732–735

Thibault, K.M., Supp, S.R., Giffin, M., White, E.P. & Ernest, S.K.M. (2011). Species composition and abundance of mammalian communities. *Ecology*, 92, 2316-2316.

U.S. Department of Agriculture, F.S. (2010). Forest inventory and analysis national core field guide (Phase 2 and 3), version 4.0. Washington, DC: U.S. Department of Agriculture Forest Service, Forest Inventory and Analysis.

White, E. P., K. M. Thibault and X. Xiao. 2012. Characterizing species abundance distributions across taxa and ecosystems using a simple maximum entropy model. Ecology, 93, 1772–1778.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

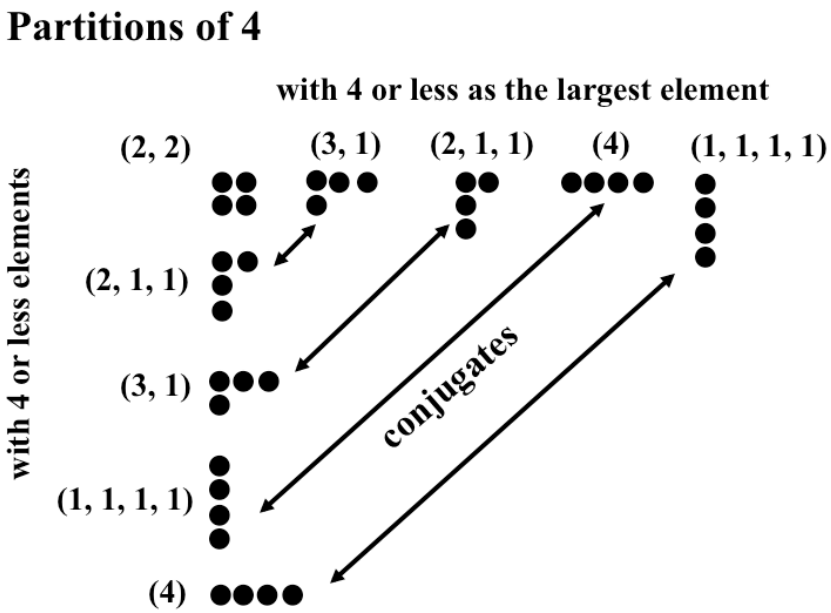

Figure 1. A configuration of Ferrers diagrams for the feasible set of 4 revealing relationship (1): that the number and set of partitions for 4 having 4, 3, 2 or less elements (vertical) equals the number and set of partitions of 4 having 4, 3, 2 or less as the largest element (horizontal). (2, 2) is a self-conjugate.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

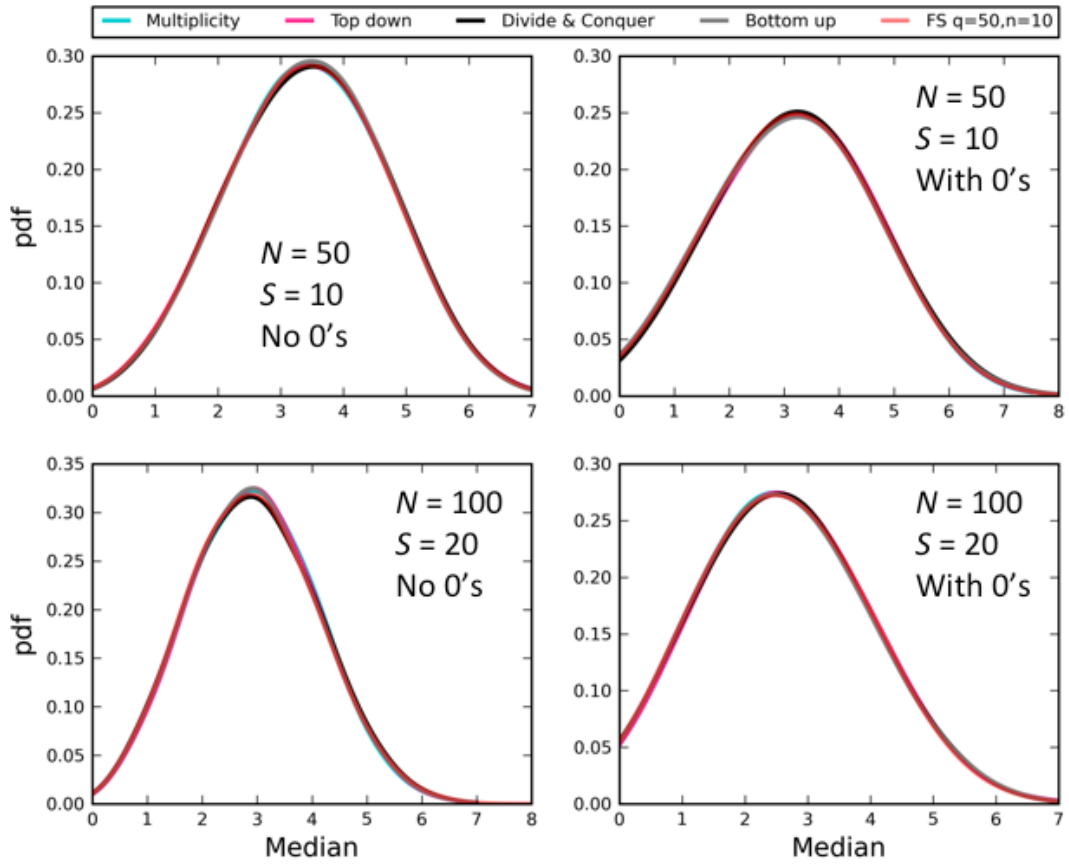

Figure 2. A comparison of the full feasible set and kernel density curves for the median derived from 1000 random samples for different combinations of N and S using our four new algorithms for parts without zeros (left column) and for parts with zeros (right column). The similarity between the results derived using our algorithms and the full feasible set reveals that the algorithms produce unbiased random samples of the feasible set. We used Sage to generate the entire feasible set for $N = 50$ and $S = 10$ (16928 partitions) and used the random partitioning function in Sage to generate 1000 partitions for $N = 100$ and $S = 20$, which is too large to enumerate in full in reasonable time (10474462 partitions).

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

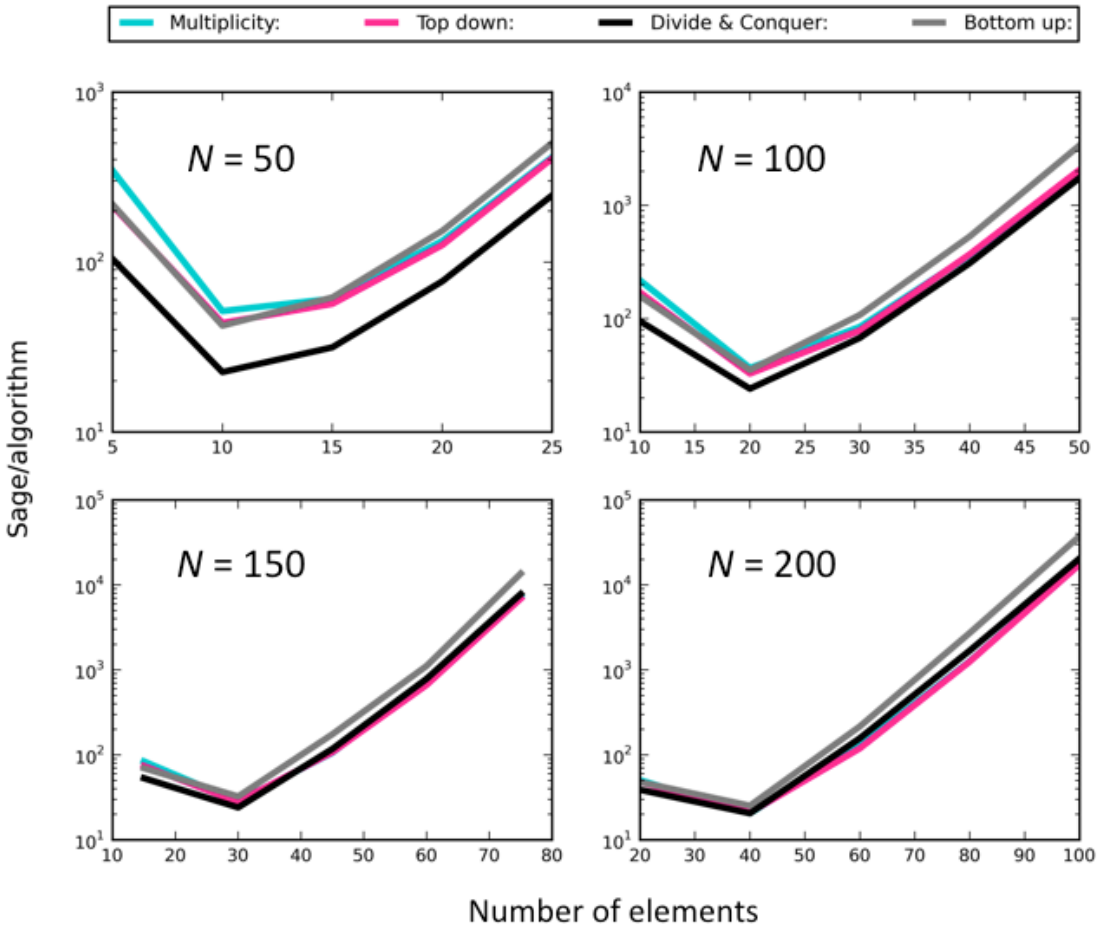

Figure 3. Plots of the ratio of the computational time for Sage to generate 300 random integer partitions (no zeros) to the time taken for the new algorithms ('Multiplicity', 'Top down', 'Divide and Conquer', 'Bottom up') to do the same. The y-axis gives the orders of magnitude

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

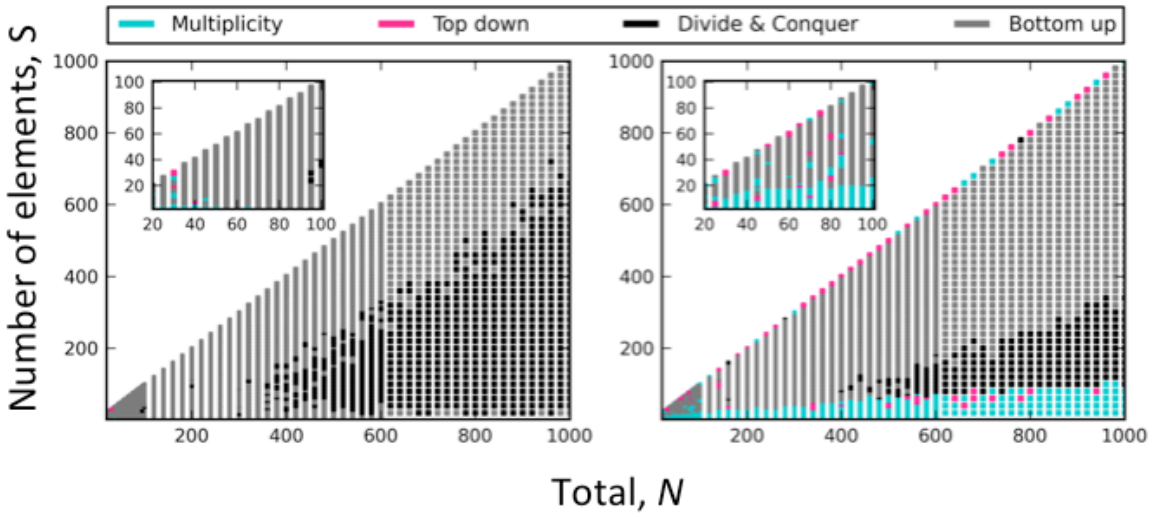

Figure 4. Color map revealing the fastest algorithm for specific combinations of $N \leq 1000$ and $S \leq N$. Comparisons were based on the time taken to generate 300 random partitions for each combination of N and S, both for cases where parts were allowed to have zero values (left) and when parts had positive values only (right). Insets reveal the small corner of the main graph where $N \leq 100$.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014

Dataset	*Number of SADs	Locey & White **>10K hours**	Present Study **~1K hours**
North American Breeding Bird Survey	2769	1586, 57%	2769, 100%
Christmas Bird Count	1992	129, 6.5%	1231, 62%
Gentry's forest transects	222	182, 82%	221, 99.5%
Forest Inventory and Analysis	10356	7359, 71%	10101, 98%
Mammal Community Database	103	42, 41%	103, 100%
Aquatic metagenomes	252	48, 19%	120, 48%
Terrestrial metagenomes	128	92, 72%	113, 88%
Fungi metagenomes	128	124, 97%	128, 100%
Total	15950	9562, **60%**	14786, **92.7%**

* *The number of SADs in the dataset matching the criteria of: one randomly chosen SAD per site having 10 or more species recorded.*

Table 1. Results of Locey and White (2013) and those from the reanalysis of those data used in that study reveal the practical gains of the algorithms developed here. Note, sampling effort per dataset was not controlled or accounted for.

PeerJ PrePrints | http://dx.doi.org/10.7287/peerj.preprints.78v2 | CC-BY 3.0 Open Access | rec: 30 Aug 2014, publ: 30 Aug 2014