**Short title**: SAMPLING FEASIBLE SETS OF ECOLOGICAL PATTERNS

1. **Article title**: Efficient algorithms forsdjoafhasojfasdfio[vhfsa[ohasfgojh sampling feasible sets of ecological patterns of abundance
2. Kenneth J. Locey‡1 and Daniel J. McGlinn\*1,2
3. Department of Biology, Utah State University, Logan, UT, 84322
4. \*Ecology Center, Utah State University, Logan, UT, 84322
5. ‡To whom correspondence should be addressed

7 1 ken@weecology.org; phone: (435) 764-5070, fax (435) 797-1575

8 2 daniel.mcglinn@usu.edu;

9

1. **Statement of authorship**: KL derived the integer partitioning algorithms, coded them in Python,
2. conducted the analyses, and served as primary author. DM coded algorithms into R, and served
3. as secondary author. Both KL and DM contributed substantially to the organization of the paper,
4. its ideas, and discussion.
5. **Keywords**: combinatorics, constraints, feasible set, integer partitions, macroecology, sampling
6. algorithms, species abundance distribution, SAD, spatial abundance distribution

16

# Words in the abstract: 269

1. **Words in the document**: 6,396

# Number of cited references: 29

1. **Number of figures**: 5
2. **Number of tables**: 1

22

* 1. SUMMARY
  2. **1.** Ecological variables such as species richness (*S*) and total abundance (*N*) strongly influence
  3. ecological patterns. Indeed, the form of the species abundance distribution (SAD) across
  4. disparate taxa can often be explained by the majority of possible forms having the same *N* and *S*,
  5. i.e., the feasible set. The feasible set reveals whether empirical patterns are exceptional to the
  6. majority of possible forms and provides simplistic constraint-based explanations for common
  7. patterns, e.g. ubiquity of hollow-curve SADs. However, use of the feasible set has been limited

**For**

* 1. to the SAD and to using inefficient sampling algorithms.
  2. **2.** We develop the only known computational algorithms for generating definitively unbiased

**Review**

* 1. random samples of the feasible set for the SAD based on both *N* and *S*. We provide Python and R
  2. based implementations and extend the feasible set approach and our algorithms to the sampling
  3. and exploration of the intraspecific spatial abundance distribution (SSAD) and other discrete
  4. distributions of wealth and abundance.
  5. **3.** The algorithms we develop are often several orders of magnitude faster than the most recently

**Only**

* 1. used approach for generating random samples of the SAD feasible set. This greatly increases the
  2. size and diversity of communities that can be examined. Defining and examining the SSAD
  3. feasible set reveals how the forms of the SAD and SSAD are coupled and reveals a potential
  4. constraint-based explanation for a general ecological phenomenon known as Taylor’s law.
  5. **4.** The algorithms we developed greatly improve the practical use of the feasible set approach,
  6. and hence, the ability of ecologists to understand and quantify how distributions of abundance
  7. among species and across space are constrained by general variables.

22

1. INTRODUCTION
2. Understanding patterns of abundance, distribution, and diversity is a primary goal in
3. ecology (Brown 1995; Rosenzweig 2002; Blackburn & Gaston 2003). These patterns are
4. frequently used to characterize community structure and provide insights into mechanisms that
5. shape communities. Additionally, the ability to predict the forms of ecological patterns has
6. served as the primary basis for evaluating ecologies theories for decades (e.g. MacArthur &
7. Wilson 1967; Scudo & Ziegler 1978; May 1981; Hubbell 2001; Harte 2011). Ecological theories
8. are underpinned by a host of different processes and mechanisms, e.g., colonization and
9. dispersal limitation, niche preemption and differentiation, stochastic population dynamics and
10. ecological equivalence, and random placement. However, it is increasingly been shown how
11. general ecological variables such as total community abundance (*N*) and species richness (*S*)
12. also have strong constraining influences on the forms of ecological patterns (Harte *et al*. 2008;
13. McGlinn & Hurlbert 2012; Supp et al. 2012; White *et al*. 2012; Locey & White 2013).
14. More than 90% of observed variation in the species abundance distribution (SAD), i.e. the
15. vector of abundances of species in a community (McGill *et al*. 2007), can be explained by
16. models constrained by *N* and *S* (Harte 2011; White *et al*. 2012). This is not surprising given that
17. the majority of possible forms of the SAD having the same *N* and *S* often explain the majority of
18. variation in abundance among species, as revealed by Locey & White (2013) who examined the
19. SAD feasible set, i.e. the set of all SADs with the same *N* and *S*. The SAD feasible set reveals
20. how *N* and *S* necessarily constrain observable variation and whether empirical SADs are
21. exceptional to or representative of the majority of possible forms (Haegeman & Loreau 2008;
22. Locey & White 2013). Likewise, the general properties of the feasible set reveal simple
23. explanations for common patterns. For example, the ubiquity of right-skewed SADs in which
24. many species have low abundance is predicted by the vast majority of possible shapes of the
    1. feasible set for ecologically realistic values of the *N* and *S*. Though only the SAD has been
    2. examined using a feasible set approach, Locey & White (2013) suggest that other ecological
    3. patterns should also be examined in the context of their feasible sets. However, there are several
    4. obstacles to extending the usage of feasible sets.
    5. Feasible sets can be immense and enumerating them can be untenable. However, small
    6. random samples can be used to characterize the center of the feasible set (i.e., average form) as
    7. well as the distribution of statistical features (e.g. evenness, diversity) within it. Locey & White
    8. (2013) took a conceptually simple and unbiased approach to sampling the SAD feasible set, an
    9. approach known as integer partitioning. This approach is based on the fact that there are a
    10. limited number of unordered ways that *n* integers can sum to a total *q*, and hence, a limited
    11. number of ways that the abundances of *S* unlabeled species can sum to a total abundance of *N*.
    12. These unordered configurations of integers are called integer partitions (Bóna 2006). For
    13. example, the feasible set for *q* = 6 and *n* = 3 is: {(4, 1, 1), (3, 2, 1), (2, 2, 2)}, where differently
    14. ordered configurations having the same integer values (e.g. (4, 1, 1), (1, 1, 4), (1, 4, 1)) represent
    15. the same integer partition, i.e. (4, 1, 1). Consequently, each integer partition represents a unique
    16. rank-abundance and frequency distributions.
    17. Although, use of integer partitioning to randomly sample the feasible set allows the
    18. feasible space to be characterized without generating all possible forms, the current sampling
    19. algorithms are computationally inefficient. All published partitioning algorithms sample the
    20. feasible set only with regards to the total, *q*. In this way, all partitions of *q* have the same
    21. probability of being drawn, regardless of the number of elements *n*. This means that randomly
    22. sampling the feasible set for a given *q* and *n*, requires generating partitions according to *q* and
    23. then rejecting those not having *n* elements, often resulting in impractically high rejection rates.
25. For example, randomly generating one partition for *q* = 1000 and *n* = 10 requires drawing from
26. a feasible set of nearly 2.4x1031 partitions, one of the roughly 8.9x1014 having 10 elements; a
27. probability of nearly 3.7x10-17.
28. Another challenge in applying integer partitioning to the study of feasible sets is that some
29. ecological patterns of abundance include parts with zero values. One example is the species
30. spatial abundance distribution (SSAD) describing the frequency with which individuals of a
31. single species occupy areas within a landscape (Brown *et al*. 1995; Harte *et al*. 2008; Haegeman
32. & Etienne 2010; Harte 2011). The SSAD reflects the spatially implicit pattern of aggregation
33. across a landscape and is mechanistically linked to other ecological patterns (Brown *et al*. 1995;
34. Harte 2011). In the SSAD, individuals can be absent from a number of areas, meaning that there
35. are some areas with zero individuals of the particular species. Because integer partitions *per se*
36. do not include zeros, integer partitioning methods need to be modified to examine the SSAD
37. feasible set.
38. Here, we present algorithms that greatly increase the efficiency of sampling the SAD and
39. SSAD feasible sets. We explain each algorithm and develop Python and R based
40. implementations of them. We test each algorithm for sampling bias and for speed against the
41. method of Locey & White (2013). To reveal the practical gains of these new algorithms, we
42. reanalyze the SAD datasets of Locey & White (2013), one of the largest and most diverse
43. compilations of species abundance data, wherein it took more than 10000 compute hours to
44. examine 60% of the available data (9562 of 15950 SADs). Our algorithms will allow us to
45. examine a larger portion of the data in less time. Finally, we examine general characteristics of
46. the SSAD feasible set, which provides a constraint-based explanation for previous connections
47. made between the distribution of abundances among species (i.e. SAD) and the distribution of
    1. individuals of a species across the landscape (i.e. SSAD) and suggest a constraint-based
    2. explanation for Taylor’s Law, i.e., the increase in variance with increase in average abundance.
    3. Our work expands the feasible set approach to an additional ecological pattern, to values of *q*
    4. (e.g., total community abundance, total species abundance) and *n* (e.g. species richness, number
    5. of samples or quadrats) that were previously untenable. This work greatly advances the front of
    6. numerical constraint-based methods in ecology (Pueyo *et al*. 2007; Haegeman & Loreau 2008;
    7. Harte *et al*. 2008; Haegeman & Etienne 2010; Harte 2011; Locey & White 2013).

8

9 METHODS

1. Here, we develop efficient and unbiased integer partitioning algorithms to generate
2. random samples of feasible sets for discrete ecological patterns such as the SAD and SSAD
3. that are defined by a total *q* and composed of *n* elements, where both *q* and *n* are positive
4. integers. Integer partitioning is a mature field of mathematics and algorithms for generating
5. random partitions of *q* (e.g. Nijenhuis & Wilf 1978) are often implemented in mathematical
6. environments (e.g. Sage, Maple, Mathematica). However, these approaches do not allow the
7. random partitioning of *q* into exactly *n* elements. Here, we use two well-established theorems
8. and a partitioning identity to develop a general method for generating a random partition of *q*
9. having *n* elements. We then modify our method to allow elements having zero values. We
10. begin by using two theorems to generate a random partition of *q* (see Appendix 1 for
11. generating functions and recurrence relations).
12. 1) For every integer *q* there are *p*(*q*) partitions having *q* or less elements.
13. 2) For every integer *q* there are *pk*(*q*) partitions having *k* or less as the first element.
14. For example, there are 5 partitions of 4 and each has 4 or less as the first element; i.e. *p*(4) = 5,
15. {(4), (3, 1), (2, 2), (2, 1, 1), (1, 1, 1, 1)}. Likewise, there are 4 partitions of 4 having 3 or less as

3 the first element; *p*3(4) = 4, {(3, 1), (2, 2), (2, 1, 1), (1, 1, 1, 1)}.

1. A random partition of *q* can be built, element by element, by iteratively applying these
2. two theorems (Fig. 1). Specifically, if we choose a random number *x* from 1 to *p*(*q*), we can
3. say there are at least *x* partitions of *q* having some value *k* or less as the first element, i.e. *x* ≤
4. *pk*(*q*). Likewise, there must also be some value of *k* – 1, for which, there are less than *x*
5. partitions of *q* having *k* – 1 or less as the first element, i.e. *pk*-1(*q*) < *x*. Putting these statements
6. together, there must be a value *k* for which *pk*-1(*q*) < *x* ≤ *pk*(*q*). In this way, we can find the
7. value of the first element in one of the partitions by finding the value of *k* that satisfies *pk*-1(*q*) <
8. *x* ≤ *pk*(*q*). Having found the value of the first element, we can decrease *x* by *pk*-1(*q*) and *q* by *k*,
9. and then find the first element for this combination of smaller values. Repeating this process
10. will sequentially build the partition until *q* = 0 and the sum of the partition is equal to the
11. original value of *q* (Fig 1).
12. The above approach is similar to well-established methods of generating random
13. partitions of *q* (e.g. Nijenhuis & Wilf 1978, Stojmenovic 2008). However, our goal is to
14. generate random partitions of *q* having *n* elements. For this, we use a well-known integer
15. partitioning identity to restrict the number of elements in a randomly chosen partition to *n*.
16. Specifically, the number of partitions of *q* having *n* elements equals the number of partitions of
17. *q* having *n* as the first element (Bóna 2006). This is because each partition of *q* having *n*
18. elements corresponds to one unique partition of *q* having *n* as the first element (Bóna 2006).
19. For example, consider the partition (3, 1), which can be illustrated with rows of dots, called a
20. Ferrer’s diagram (Fig 1). In the Ferrer’s diagram for (3, 1) there are two rows, the largest
    1. having three dots. Flipping the diagram on its diagonal produces its conjugate (2, 1, 1), which
    2. has three rows, the largest row having two dots (Fig 1). So, the conjugate of (3, 1) is (2, 1, 1)
    3. and vice versa. Consequently, the first part of an integer partition determines the number of
    4. parts in its conjugate (Bóna 2006). This allows us to extend the problem of generating random
    5. partitions of *q* to random partitions of *q* having exactly *n* elements. That is, knowing the first
    6. element must be *n* so that its conjugate has *n* elements, we can decrease *q* by *n* and then
    7. generate a random partition for this decreased value of *q* having *n* or less as the first element.
    8. Once the partition is generated, we append *n* to the beginning of the partition and conjugate it
    9. to produce a random partition of the original *q* having exactly *n* elements (Fig 1).
    10. The approach outlined above begins with a randomly chosen number *x* between 1 and *p*(*q*).
    11. It then finds the value of *k* that satisfies the inequality *pk*-1(*q*) < *x* ≤ *pk*(*q*), which is the value of
    12. the first element of the partition*.* However, the question remains as to which value of *k* to start
    13. with and how to proceed to different values. Indeed, we could start with the smallest possible
    14. value of *k* (*k* = 1) and take a ‘bottom-up’ approach, or the largest possible value and take a ‘top-
    15. down’ approach, or even choose *k* at random and use a ‘divide-and-conquer’ method. These
    16. approaches differ only in how *k* is chosen and each builds the partition one element at a time.
    17. However, in the event that *q* is much larger than *n*, e.g. all trees in the 50 ha Barro-Colorado
    18. Island mapped forest plot where *q* ≈ 200,000 individuals and *n* ≈ 300 species, the three
    19. algorithms above will still be inefficient. This is because they would first generate a partition of,
    20. say, 200000 having 300 as the first element, but having as many as 199701 elements, and then
    21. conjugate it to produce a partition of 200000 having 300 elements. Clearly building the partition
    22. one element at a time would be inefficient in this case.
21. An alternative approach is to build a partition using multiples of integers – the
22. ‘multiplicity’ approach. Instead of finding the value of *k* corresponding to the first element,
23. appending it, and moving on, we can instead ask how many times must *k* occur, i.e. the partitions
24. of *q* having some multiple *m* of *k*. We can start with the smallest possible multiple (i.e. *m* = 1)
25. and ask whether *x* is less than or equal to the number of partitions of *q* – *k*\**m* having less than *k*
26. as the first part. This is because the set of partitions of *q* having a number of *k*’s equal to *m*
27. actually contains the set of partitions of *q* – *k*\**m* having less than *k* as the first part (Appendix 1).
28. We can increase *m* by one until *x* ≤ *pk*(*q* – *k*\**m*), at which point we will have found the
29. corresponding multiple of *k.* One drawback to this method is the time taken for the extra
30. computation.

11

1. *Random partitions for q and n, with some parts having zero values*
2. The above algorithms address distributions having positive values, such as the distribution
3. of abundance among species (SAD). In contrast, some ecological patterns include zero values
4. (e.g. absences). One example is the species spatial abundance distribution (SSAD), a frequency
5. distribution that characterizes the number of quadrats, cells, or areas containing a given
6. abundance of a species (Brown *et al*. 1995; Haegeman & Etienne 2010; Harte 2011). However,
7. only small changes are needed to adapt the above approaches to cases allowing zero-valued
8. parts. For example, let 10 unlabeled individuals occupy a landscape sectioned into quarters. The
9. most aggregated distribution would be for all 10 to occupy the same quarter, [10, 0, 0, 0]. The
10. least aggregated would be for 3 to occupy two quarters while 2 occupy the other two quarters, [3,
11. 3, 2, 2]. In fact, the number of configurations for 10 unlabeled individuals distributed across 4
12. unlabeled sections equals the number of partitions of 10 having 4 or less parts, i.e. *p*4(10 + 4) =
    1. 23. Consequently, if *n* ≤ *q*, a random partition for *q* and *n* allowing for zero-valued parts, is
    2. simply a random partition for *q* having *n* or less parts, with zeros appended to ensure the final
    3. form of the partition has *n* parts.
    4. On the other hand if *n* > *q* a different approach is needed. To see this let 4 unlabeled
    5. individuals occupy a landscape sectioned into tenths. The most aggregated distribution would be
    6. for all 4 to occupy the same subsection, [4, 0, 0, 0, 0, 0, 0, 0, 0, 0] and the least aggregated
    7. configuration would be for 4 sections to have one individual and for 6 sections to have zero, i.e.
    8. [1, 1, 1, 1, 0, 0, 0, 0, 0, 0]. In this way, the number of possible configurations for 4 unlabeled
    9. individuals distributed across 10 unlabeled sections is *p*(*q*). Consequently, if *q* < *n*, a random
    10. partition for *q* and *n*, allowing for zero-valued parts, is simply a random partition for *q* having *q*
    11. or less parts, with zeros appended to ensure the partition has *n* parts.

12

1. *Examining for bias and speed*
2. We implemented the above algorithms in Python and R and made them freely available
3. using a public Github repository (https://github.com/klocey/partitions). We also developed these
4. algorithms into an R package (“rpartitions”) to be distributed on The Comprehensive R Archive
5. Network (CRAN; [http://cran.us.r-project.org/).](http://cran.us.r-project.org/)) We used kernel density curves to visually
6. compare the results of the above algorithms to full feasible sets and random samples generated
7. with the function implemented in the Sage mathematical environment (www.sagemath.org) that
8. is based on the algorithm of Nijenhuis and Wilf (1978) and is the method used in Locey & White
9. (2013). If our algorithms are unbiased, then their distributions will not differ in any systematic
10. way from full feasible sets and random samples generated using the proven function
11. implemented in Sage.
12. Additionally, our Python and R packages include test files that conduct, among several
13. other tests of our source code and partitioning functions, 2-sample t-tests and 2-sample
14. Kolmogorov Smirnov tests on kernel density curves of the variance of logarithmically
15. transformed abundances from random samples generated by the Sage software and random
16. samples generated using the algorithms derived here. These tests are important to regularly run
17. because additional code developments can corrupt source code.
18. We compare the computational speed of our algorithms to that of the approach used in
19. Locey & White (2013) (i.e., using Sage to generate random partitions for a given *q* and rejecting
20. those not having *n* elements) across a range of values of *q*, *n*, and *q*-*n* ratios for which the latter
21. method was likely to return random samples within reasonable time (one hour). Because Sage is
22. coded in Python, our comparisons are made using the Python versions of our algorithms.

12

1. *Empirical Demonstration of the New Algorithms*
2. Locey & White (2013) analyzed the species abundance distributions (SADs) of 9562 sites
3. of trees, bird, mammal, fungi, and prokaryote communities using a partitioning algorithm that
4. sampled the feasible set according to total abundance *N* but not with respect to species richness *S*
5. (i.e. the number of elements). Those data consisted, in part, of a subset of previously compiled
6. datasets of site-specific species abundance data (see White *et al*. 2012), and included four
7. continental-to-global scale surveys, including the Christmas Bird Count (129 sites) (National
8. Audubon Society 2002), North American Breeding Bird Survey (1,586 sites) (Sauer *et al*. 2011),
9. Gentry’s Forest Transect Data Set (182 sites) (Phillips & Miller 2002), Forest Inventory Analysis
10. (7,359 sites) (U.S. Department of Agriculture 2010), and one global-scale data compilation, the
11. Mammal Community Database (42 sites) (Thibault *et al*. 2011). Locey & White (2013) also
    1. compiled abundance data at the species level from five microbial metagenome projects for a total
    2. of 264 SADs. Those data were obtained from the metagenomics server MG-RAST (Meyer et al.
    3. 2008). Metagenomic data were compiled into datasets representing aquatic prokaryotic
    4. communities (48 metagenomes) (Flores *et al*. 2011,
    5. [www.catlin.com/en/Responsibility/CatlinArcticSurvey),](http://www.catlin.com/en/Responsibility/CatlinArcticSurvey)) terrestrial prokaryotic communities (92
    6. metagenomes) (Chu et al. 2010; Fierer *et al*. 2012), and terrestrial fungal communities (124
    7. metagenomes) (Amend *et al*. 2010). We refer the reader to Locey & White (2013) and White *et*
    8. *al*. (2012) for more thorough descriptions of those datasets.
    9. The inefficiency of the partitioning method used by Locey & White (2013) restricted their
    10. analyses to combinations of abundance *N* and species richness *S*, for which, there was a
    11. reasonable probability of generating a random integer partition of *N* with exactly *S* elements.
    12. This restriction allowed for only 60% of the available data to be examined despite more than
    13. 10000 compute hours worth of effort. We reanalyze those datasets using the algorithms
    14. developed here, which should allow for random samples of a greater number of SADs to be
    15. produced in less time.

16

1. *General characteristics of the SSAD feasible set*
2. Brown *et al*. (1995) revealed evidence that the general form of the SSAD, like that of the
3. SAD, is characterized by a hollow-curve. In the sense of the SSAD, a hollow-curve implies that
4. many areas are occupied by few or no individuals and that relatively few areas are occupied by
5. many individuals. Both Brown *et al*. (1995) and Harte (2011) mechanistically couple the SSAD
6. and SAD using arguments based on local niche differences and the constraint-based approach of
7. entropy maximization, respectively. We generated random samples of the feasible set of the
8. SSAD for ecologically realistic combinations of *q* and *n*, and examined their general features.

3

1. RESULTS
2. Statistical properties of entire feasible sets are indistinguishable from random samples
3. generated with our sampling algorithms, demonstrating that the implementations of our
4. algorithms were unbiased (Fig 2 and Figs 1-2 of Appendix). When generating 300 random
5. partitions, i.e. enough to safely characterize the feasible space (Locey & White 2013), these
6. implementations were, at worst, 10 to 100 times faster than the method used by Locey & White
7. (2013) and were, at best, 10000 to 100000 times faster for the combinations of *q* and *n* we tested
8. (Fig 3). These combinations were limited to values for which the algorithm used in Sage could
9. generate random samples in reasonable time. Consequently, the algorithms we developed
10. quickly produce random samples for values of *q* and *n* that are impractical with algorithms that
11. sample only according to *q*. Each algorithm was best suited for particular values of *q* and *n* (Fig
12. 4). For cases where all parts have positive values, the multiplicity algorithm is the fastest for
13. combinations where *q* is partitioned among a relatively small number of elements (Fig 3
14. Appendix).
15. The greater efficiency of the algorithms developed here allowed us to generate between
16. 300 and 500 random partitions for 92.7% of the SADs (14786/15950) from the compilation of
17. SAD data used by Locey & White (2013), in less than 1000 compute hours. In contrast, the
18. method used by Locey & White (2013) required more than 10000 compute hours to generate
19. between 300 and 500 random partitions for 60% of the available data (9562/15950 SADs).
20. Our examination of the SSAD feasible set supports the observation of Brown *et al*. (1995)
21. that SSADs are characterized by hollow-curves (few cells with many individuals and many cells
    1. with few individuals) (Fig 5). The hollow-curve nature of the SSAD feasible set increases as the
    2. total *q* is distributed across a greater number of elements (e.g. quadrats, subplots, cells). This
    3. reveals an expectation for decreased variation in the spatially implicit distribution of a species
    4. (i.e., most samples are going to have few individuals and few samples will have many) as *q* is
    5. distributed across an increasing number of elements. According to the feasible set, variance
    6. increases with increasing average cell abundance.

7

1. DISCUSSION
2. The feasible set approach based on integer partitioning is a contextual framework for
3. understanding how constraints influence the forms of discrete ecological patterns and in discrete
4. distributions of wealth, size, and abundance (Locey & White 2013). We used integer partitioning
5. to understand how the feasible set is ordered, to find the size and general features of the feasible
6. set, and to generate unbiased random samples of the feasible set for a given a total (e.g. total
7. community abundance, total species abundance) and number of entities (e.g. species, quadrats).
8. The algorithms we derived greatly increase the practical use of feasible set by decreasing
9. computing time. In addition to examining the distribution of abundance among species (i.e.
10. SAD), we expanded the feasible set approach to discrete distributions with zero values, such as
11. the spatially implicit distribution of individuals of a species within a landscape (i.e. SSAD). We
12. also provided the algorithms in two computing languages frequently used by ecologists, R and
13. Python, and have taken steps to ensure our implementations are unbiased and have provided a
14. script to detect computational errors that can result from future code development (e.g. when a
15. user modifies the code).
16. Integer partitioning is only one way to examine and randomly sample the feasible set of
17. possible SAD and SSAD shapes. Other possibilities include constraint-based programming (see
18. <http://cran.r-project.org/web/views/Optimization.html)>and iterative random walks, such as that
19. used by Haegeman & Loreau (2008). Those approaches may not require combinatorial problems
20. to be solved and so may not suffer from the problem of combinatorial explosion (large increases
21. in the size of the feasible set for small changes in the total q and number of elements). However,
22. as stated by Locey & White (2013) one benefit to the integer partitioning approach is that the
23. random sampling algorithms are inherently unbiased and do not require ‘burn-in’ periods to
24. produce effectively independent samples. The combinatorial approach also reveals properties
25. such as the size of the feasible set, the distribution of statistical characteristics (e.g. species
26. evenness, diversity, modal abundance class) across the feasible set, and connections between
27. different ecological patterns, e.g. SAD and SSAD. However, we suggest that constraint-based
28. programming and iterative random walks should also be examined and compared to the
29. combinatorial approach.
30. Our examination of the SSAD feasible set (Fig. 5) reveals that the central tendency of the
31. set is characterized by a hollow-curve which is consistent with the empirical SSADs found by
32. Brown *et al*. (1995). In that study, the authors state that the highly ‘clumped’ and hollow-curve
33. nature of the SSAD resembles distributions used to predict the form of the SAD. The authors
34. offer an ecological interpretation for the similarity between the patterns in terms of niche
35. requirements. Likewise, the Maximum Entropy Theory of Ecology (Harte 2011) also argues for
36. a fundamental but purely constraint-based relationship between the forms of the SAD and SSAD.
37. However, in terms of their feasible sets determined by the total and the number of entities, the
38. SSAD differs from the SAD only in that zero values are allowed. Consequently, the forms of the
    1. two patterns are not only coupled by ecological and statistical mechanisms, but are more simply
    2. coupled by the mathematical properties of their feasible sets.
    3. The results of examining general properties of the SSAD feasible set suggest the potential
    4. for a simple constraint-based explanation for the increase in variance with average abundance,
    5. i.e. Taylor's Law (Taylor 1961). Taylor’s Law also generally predicts a linear slope between 1
    6. and 2 for the log-log relationship of the variance versus mean abundance (Kilpatrick & Ives
    7. 2003). If a feasible set based examination of this relationship consistently yields slopes between
    8. 1 and 2, then this would provide a second feasible set based explanation for a pattern that has
    9. been intensively studied in ecology but commonly observed in other disciplines; the ubiquity of
    10. the hollow-curve frequency distribution being the other.
    11. The feasible set approach taken here and in Haegeman & Loreau (2008) and in Locey &
    12. White (2013) ignores biological and statistical mechanism and focuses entirely on observable
    13. variation in the shape of empirical patterns. Consistency of empirical patterns with the center of
    14. the feasible set suggests that the shapes of those patterns contain little information beyond that
    15. encoded by the constraints used to characterize the feasible set (Haegeman & Loreau 2008;
    16. Locey & White 2013). However, consistency with the feasible set does not mean that biological
    17. processes are not operating but rather that they may indirectly influence empirical patterns
    18. through their effects on constraints (Supp *et al*. 2012; White *et al*. 2012). Indeed, if the majority
    19. of variation in an ecological pattern can be explained/predicted by a few general variables, then it
    20. is probably important to the prediction of the pattern to understand the forces, processes, or
    21. mechanisms driving the values of the variables (McGill 2010). Alternatively if empirical patterns
    22. occupy an uncommon portion of the feasible set (e.g., in being exceptionally uneven) biological
39. processes or additional constraints beyond those used to characterize the feasible set may be
40. relevant.
41. Our work greatly advances the ability of ecologists to characterize and explore observable
42. variation in ecological patterns of abundance by greatly decreasing computational time and by
43. defining the feasible set of another ecological pattern of abundance, i.e., the SSAD. These
44. advances allow combinations of constraint values to be examined that were previously out of
45. reach, and hence, will help provide a greater understanding of the degree to which small
46. combinations of general variables can explain the forms of ecological distributions (e.g. SAD,
47. SSAD) and common ecological patterns (e.g. hollow-curve frequency distributions, Taylor’s
48. Law). The algorithms we developed apply to frequency distributions such as the SAD and
49. SSAD. However, many ecological patterns are also cumulative, describing the rates at which
50. species are encountered with increasing area (species-area relationship) or time (species-time
51. relationship) or both (species-time-area relationship), as well as the spatially implicit distribution
52. of occupancy among species within a landscape (occupancy-frequency distribution).
53. Characterizing and randomly sampling the feasible sets of these and other patterns may require
54. modification of the algorithms we developed, approaches more similar to that of Haegeman &
55. Loreau (2008), or altogether new approaches.

18

1. ACKNOWLEDGMENTS
2. We thank X. Xiao and E. P. White for critical discussions and friendly reviews. We thank the
3. numerous individuals involved in collecting and providing the data used in this paper including
4. the essential citizen scientists who collect the North American Breeding Bird Survey and
5. Christmas Bird Count data, USGS and CWS scientists and managers, researchers who collected
6. and sequenced the microbial metagenomic data, the MG-RAST project, the Ribosome Database
   1. Project, the Audubon Society, the U.S. Forest Service, the Missouri Botanical Garden, and
   2. Alwyn H. Gentry.

3

1. LITERATURE CITED
2. Amend, A.S., Seifert, K.A., Samson, R. & Bruns, T.D. (2010) Indoor fungal composition is
3. geographically patterned and more diverse in temperate zones than in the tropics.
4. *Proceedings of the National Academy of Sciences USA*, 107, 13748-13753.
5. Bóna, M. (2006) *A walk through combinatorics: An introduction to enumeration and graph*
6. *theory*. 2nd Edition. World Scientific Publishing Co. Singapore.
7. Brown, J. H. (1995) *Macroecology*. Univ. Chicago Press, Chicago.
8. Brown, J.H., Mehlman, D.W. & Stevens, G.C. (1995) Spatial variation in abundance. *Ecology*,

12 76, 2028-2043.

1. Chu, H., Fierer, N., Lauber, C.L., Caporaso, J.G., Knight, R. & Grogan, P. (2010) Soil bacterial
2. diversity in the Arctic is not fundamentally different from that found in other biomes.
3. *Environmental Microbiology*, 12,2998–3006.
4. Fierer, N., Lauber, C.L., Ramirez, K.S., Zaneveld, J., Bradford, M.A. & Knight, R. (2012)
5. Comparative metagenomic, phylogenetic and physiological analyses of soil microbial
6. communities across nitrogen gradients. *The ISME Journal*, 6, 1007–17.
7. Flores, G.E., Campbell, J., Kirshtein, J., Meneghin, J., Podar, M., Steinberg, J.I. *et al*. (2011)
8. Microbial community structure of hydrothermal deposits from geochemically different vent
9. fields along the Mid-Atlantic Ridge. *Environmental Microbiology*, 13, 2158-2171.
10. Haegeman, B. & Etienne, R.S. (2010) Entropy Maximization and the Spatial Distribution of
11. Species. *The American Naturalist*, 175, E74–E90.
12. Haegeman, B. & Loreau, M. (2008) Limitations of entropy maximization in ecology. *Oikos*,

1 117, 1700–1710.

1. Haegeman, B. & Loreau, M. (2009) Trivial and non‐trivial applications of entropy maximization
2. in ecology: a reply to Shipley. *Oikos*, 118, 1270-1278.
3. Harte, J., Zillio, T., Conlisk, E. & Smith, A.B. (2008) Maximum entropy and the state-variable
4. approach to macroecology. *Ecology*, 89, 2700–2711.
5. Locey, K*.*J. & White, E.P. (2013) How species richness and total abundance constrain the
6. distribution of abundance. *Ecology Letters*. 16, 1177-1185.
7. MacArthur, R.H. & Wilson, E.O. (1967) *The theory of island biogeography* (Vol. 1). Princeton
8. University Press.
9. May, R.M. (1981) Theoretical ecology. Principles and applications. *Theoretical ecology.*
10. *Principles and applications.*, (Ed. 2).
11. McGill, B.J. (2010) Towards a unification of unified theories of biodiversity. *Ecology Letters*,

13 13, 627–642.

1. McGill, B.J., Etienne, R.S., Gray, J.S., Alonso, D., Anderson, M.J., Benecha, H.K. *et al*. (2007)
2. Species abundance distributions: moving beyond single prediction theories to integration
3. within an ecological framework. *Ecology Letters*, 10, 995–1015.
4. McGlinn, D.J. & Hurlbert, A.H. (2012) Scale dependence in species turnover reflects variance in
5. species occupancy. *Ecology*, 93, 294–302.
6. Meyer, F., Paarmann, D., D’Souza, M., Olson, R., Glass, E.M., Kubal, M. *et al*. (2008) The
7. metagenomics RAST server - a public resource for the automatic phylogenetic and functional
8. analysis of metagenomes. *BMC Bioinformatics*, 9, 386.
9. National Audubon Society. (2002) *The Christmas Bird Count historical results*. Retrieved from
10. [http://www.audubon.org/bird/cbc.](http://www.audubon.org/bird/cbc)
    1. Nijenhuis, A. & Wilf, H.S. (1978) *Combinatorial Algorithms for Computers and Calculators*.
    2. Academic Press, New York.
    3. Pueyo, S., He, F. & Zillio, T. (2007) The maximum entropy formalism and the idiosyncratic
    4. theory of biodiversity. *Ecology Letters*, 10, 1017-1028.
    5. Sauer, J.R., Hines, J.E., Fallon, J.E., Parkieck, D.J., Ziolkowski, D.J. Jr. & Link, W.A. (2011)
    6. *The North American Breeding Bird Survey 1966-2009*. Version 3.23.2011. USGS Patuxent
    7. Wildlife Research Center, Laurel, MD.
    8. Scudo, F.M. & Ziegler, J.R. (1978) *The Golden age of theoretical ecology, 1923-1940: a*
    9. *collection of works by V. Volterra, VA Kostitzin, AJ Lotka, and AN Kolmogoroff*. Springer-
    10. Verlag.
    11. Stojmenovic, I. (2008) Generating all and random instances of a combinatorial object. In
    12. *Handbook of Applied Algorithms: Solving scientific, Engineering and Practical Problems*.
    13. John Wiley & Sons, Inc., New Jersey, pp. 1-38.
    14. Storch, D., Šizling, A.L., Reif, J., Polechová, J., Šizlingová, E. & Gaston, K.J. (2008) The quest
    15. for a null model for macroecological patterns: geometry of species distributions at multiple
    16. spatial scales. *Ecology Letters*, 11,771–784.
    17. Supp, S.R., Xiao, X., Ernest, S.K*.*M. & White, E.P. (2012) An experimental test of the
    18. response of macroecological patterns to altered species interactions. *Ecology*, 93, 2505–2511.
    19. Taylor, L.R. (1961) Aggregation, variance and the mean. *Nature*, 189, 732–735
    20. Thibault, K.M., Supp, S.R., Giffin, M., White, E.P. & Ernest, S.K.M. (2011). Species
    21. composition and abundance of mammalian communities. *Ecology*, 92, 2316-2316.
    22. U.S. Department of Agriculture, F.S. (2010). Forest inventory and analysis national core field
    23. guide (Phase 2 and 3), version 4.0. Washington, DC: U.S. Department of Agriculture Forest
11. Service, Forest Inventory and Analysis.
12. White, E.P., Thibault, K*.*M. & Xiao, X. (2012) Characterizing species abundance distributions
13. across taxa and ecosystems using a simple maximum entropy model. *Ecology*, 93, 1772–

4 1778.

**Theorem 1**: For every integer *q* there are *p*(*q*) partitions having *q* or less elements.

**Theorem 2**: For every integer *q* there are *pk*(*q*) partitions having *k* or less as the first element.

**Identity**: Each partition of *q* with *n* elements has a conjugate partition with *n* as the first element

**Example:** A partition for *q* = 4 having *n* = 2 elements corresponds to the conjugate of that partition which has 2 as its first element.

**(3, 1)**

**(2, 1, 1)**

**Problem 1**: Generate a random partition of *q* that has any number of elements.

**Solution**: Use theorem 1 to choose a random integer *x* from 1 to *p*(*q*). Repeatedly use theorem 2 to sequentially build the partition.

**Example:** If *q* = 9 & *x* = 21, then the first element is *k* = 5 because *p*4(9) = 18 < 21 ≤ 23 = *p*5(9) partition: **(5)** Decrease *x* by *p*4(9) & *q* by *k*

Now, *q* = 4 & *x* = 3, the first element is *k* = 2 because *p*1(4) = 1 < 3 ≤ 3 = *p*2(4) partition: **(5, 2)** Decrease *x* by *p*1(4) & *q* by *k*

Now, q = 2 & *x* = 2, the first element is *k* = 2 because *p*1(2) = 1 < 2 ≤ 2 = *p*2(2) partition: **(5, 2, 2)** Decrease *x* by *p*1(2) & *q* by *k*

**Problem 2:** Generate a random partition that sums to *q* and has *n* elements

**(5, 2, 2)**

**Solution:** Generate a random partition of *q* – *n* having *n* or less as the first element. Append *n* and find the conjugate to produce a partition of *q* with *n* elements.

**Example:** Generate a random partition for *q* = 14 having *n* = 5 elements:

1.) Generate a partition for *q* = 9 with 5 or less as the first element.

2.) Append 5 to the partition. 3.) Conjugate the partition.

1

**(5, 2, 2)**

**append *n***

**(5, 5, 2, 2)**

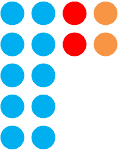
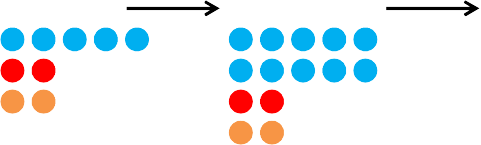
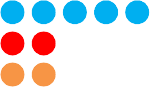
**conjugate**

**(4, 4, 2, 2, 2)**

**For**

**Review**

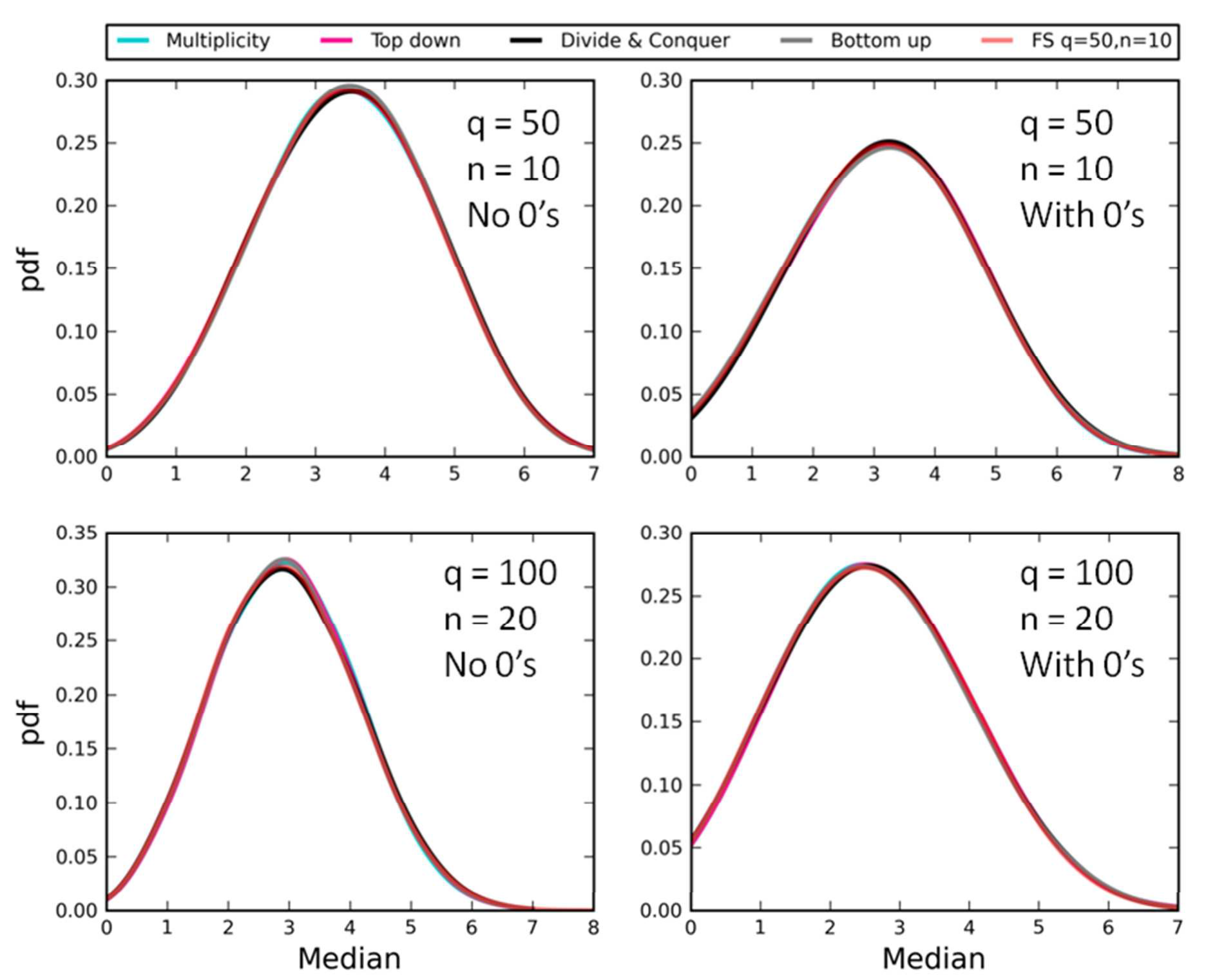
* 1. **Figure 1**. *Top:* Two theorems and one identity used to generate random integer partitions.



* 1. *Center:* General method for generating a random partition of *q* having 1 to *q* elements using the
  2. two theorems. *Bottom:* General method for generating a random partition of *q* into exactly *n*
  3. elements, using the two theorems and the partitioning identity.

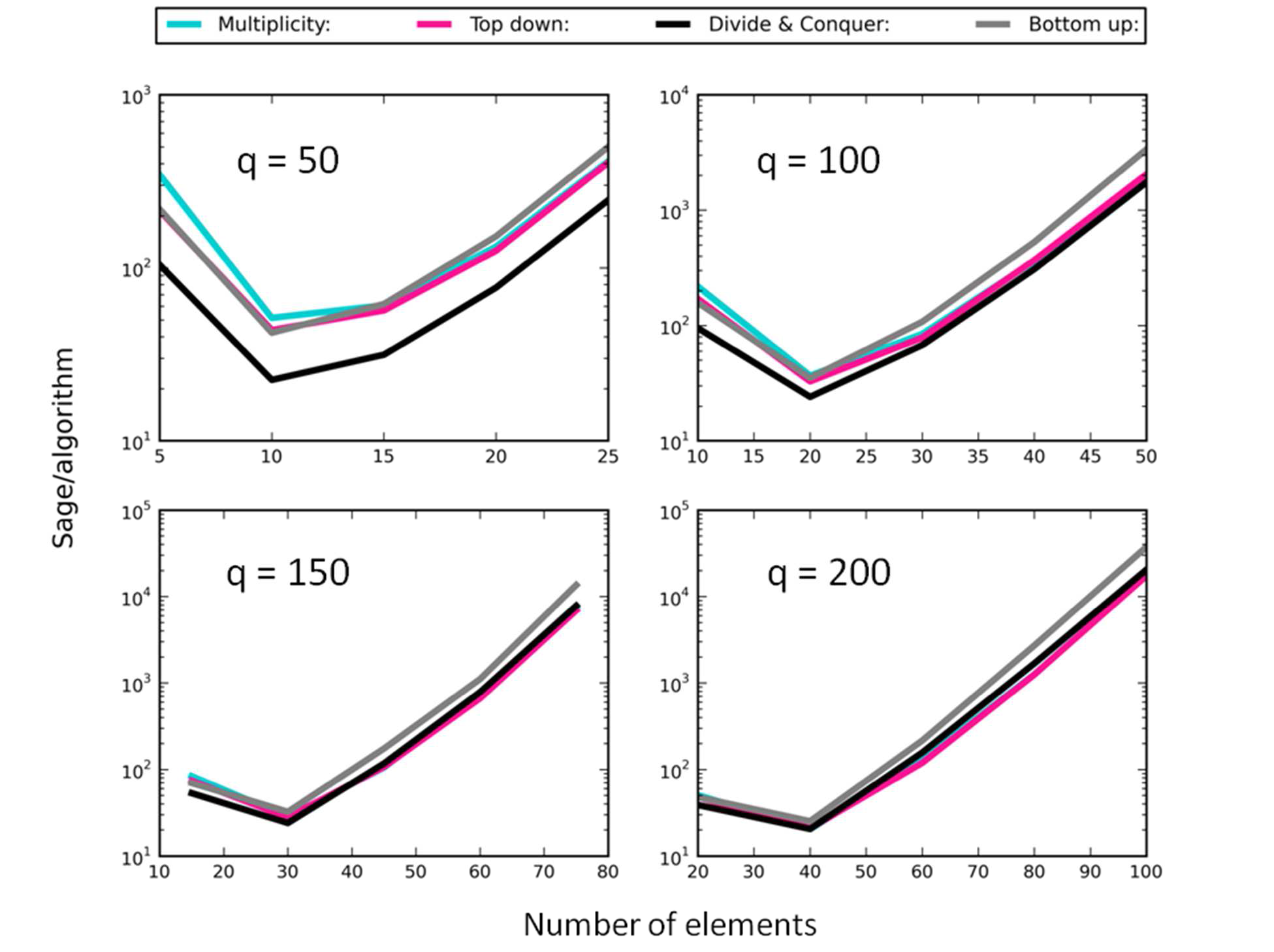
6

**For**

1

**Review**

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

1 

**For**

**Review**

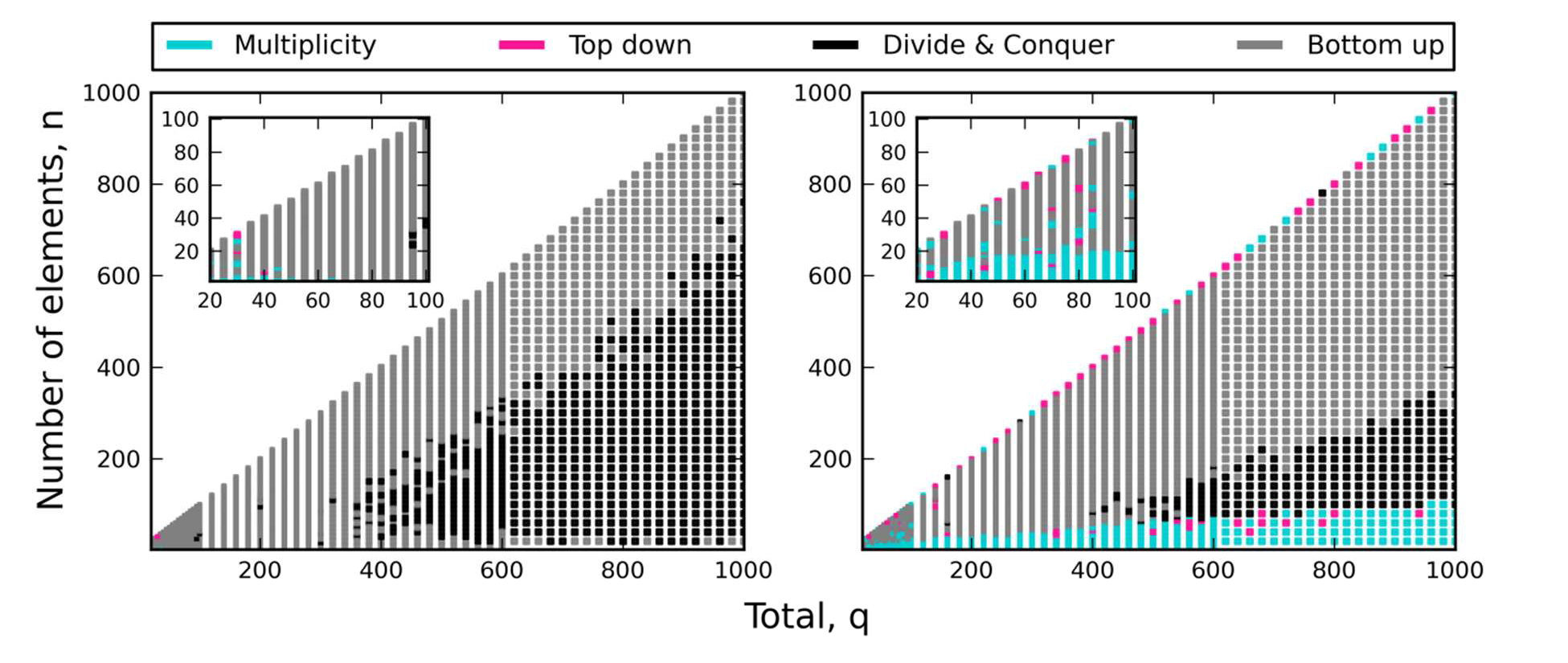
* 1. **Figure 3**. Plots of the ratio of the computational time for Sage to generate 300 random integer
  2. partitions (no zeros) to the time taken for the new algorithms (‘Multiplicity’, ‘Top down’,
  3. ‘Divide and Conquer’, ‘Bottom up’) to do the same across a range of *n* values at four different
  4. values of *q*. The ratio is plotted on a log-transformed axis.

6

7

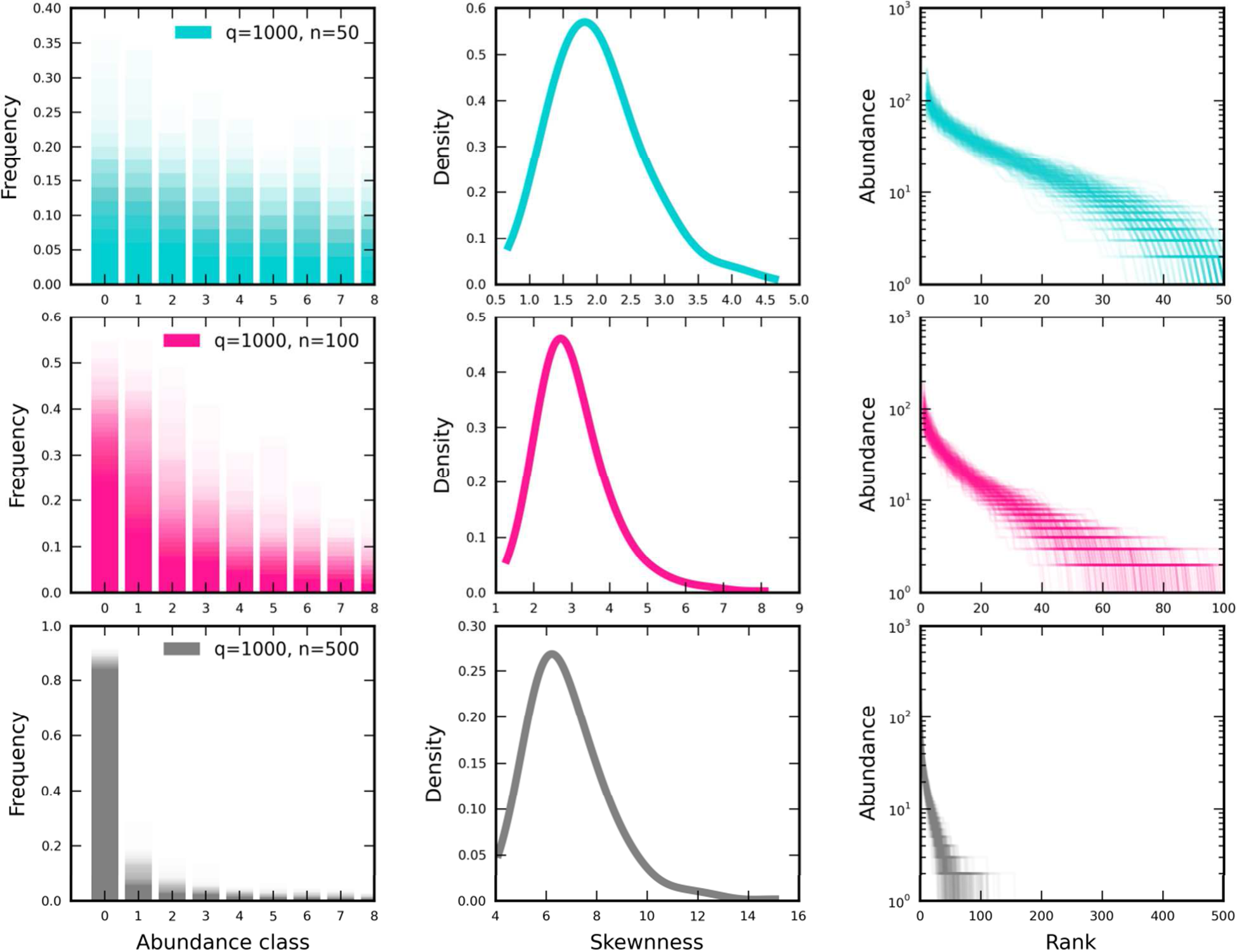
8

9

1

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

7

1 

**For**

**Review**

2

* 1. **Figure 5**. The SSAD feasible sets for *q* = 1000 and *n* = {50, 100, 500} for 300 random samples
  2. displayed as frequency distributions (left), kernel density curve of skewness (center), and ranked
  3. distributions of abundance (i.e. areas or subplots ranked from most-to-least occupied) (right).
  4. Each row applies to one value of *n*. For n = 500, ranks with values of 0 are plotted but difficult to
  5. discern.

8

|  |  |  |  |
| --- | --- | --- | --- |
| 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%** |

1. \* The number of SADs in the dataset matching the sampling criteria (see methods).
2. **Table 1**. The number and proportion of the SAD datasets that we analyzed in the present study
3. compared to the past analysis of Locey & White (2013).Note, sampling effort per dataset was not
4. controlled or accounted for.