

**Realizing the Niche's Breadth:
Inferring Ecological Process with Species Generalism**

Jasper McChesney

A thesis submitted to the faculty of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements for the degree of Masters of Science in
Ecology.

March 2010

Chapel Hill

Advisory Committee

Robert K. Peet

Peter S. White

Alan S. Weakley

Abstract

Jasper McChesney: Realizing the Niche's Breadth:
Inferring Ecological Process with Species Generalism
(Under the direction of Robert K. Peet)

Various ecological processes affect generalist and specialist species differently. By measuring niche breadth in communities, we can infer those processes. Simulation models provided proofs of concept for three analyses that were applied to Carolina Vegetation Survey data. (1) A refined method for estimating niche breadth using co-occurrence data is presented. (2) It is shown that sampling grain can affect the quality of such estimates and also illuminate the scale of the processes limiting species' distributions. Tress showed surprising sensitivity to spatial variation at a small scale (10 cm²) while herbs responded more strongly at a larger one (1000 m²). (3) The relationship between assemblage richness and the generalism of its members is explored. There is evidence that in some community types, important factors affecting richness include an evolutionary generalism-competition trade-off and the availability of specialist competitors in the local pool.

Table of Contents

Abstract	ii
List of Tables	iv
List of Figures.....	v
Acknowledgments	vii
Preface.....	viii
1. Variations in Theta.....	1
2. The Multi-Grain Niche	41
3. A Wealth of Niches	73
Postscript.....	100
Appendices	102

List of Tables

Table 1-1. Simulation parameters were varied to create 72 distinct scenarios.....	10
Table 1-2. Performance selected metrics.....	22
Table. 1-3. Woody plants in the CVS database	25
Table 2-1. Simulation parameters.....	47
Table 3-1. Fixed simulation parameters.....	80
Table 3-2. Parameters varying by scenario.....	80
Table 3-3. Relationships between four per-plot generalism statistics and plot richness	84
Table 3-4. Dependence of plot generalism on richness in all of CVS.	86
Table 3-5. Communities in CVS matching each of the proposed models.	89

List of Figures

Fig. 1-1. Example niche curves and three possible focal species widths (a-c).	5
Fig. 1-2 Effect of niche width means and distribution on estimates (r^2).	12
Fig. 1-3. Positioning of sites (abscissa) and niche optima (panels) on a gradient.	13
Fig. 1-4. Increase in ranked metric performance by sub-sampling	15
Fig. 1-5. Rank performance of three co-occurrence counting methods	16
Fig. 1-6. Interaction of niche sizes with co-occurrence counting methods.	18
Fig. 1-7. Effect of plots saturation on metric rank for four richness correction methods.	20
Fig. 1-8. Performance of selected metrics with various mean widths	23
Fig. 1-9. Scaled commonness and niche width estimates from metric #55n of CVS woody plants.	24
Table 2-1. Simulation parameters.	47
Fig. 2-2. Performance with different sampling grains.	50
Fig. 2-3. Causes of plot heterogeneity	51
Fig. 2-4. Effect of plot heterogeneity on metric performance.	52
Fig. 2-5. Most suited grain for measuring niche widths at the given level of spatial heterogeneity	53
Fig. 2-6. Density plot showing the difference, in area, between the best performing grain in each scenario and the grain having the greatest variation in its species' niche width estimates..	55
Fig. 2-7. Change in mean generalism score as sampling grain varies.....	56
Fig. 2-8. Grain correction techniques..	58
Fig. 2-9. Generalism scores for all sampled species in the Carolina Vegetation Survey data.	61
Fig. 2-10. Distribution of species' most limiting scale.....	62
Fig. 2-11. Selected species illustrating different relationships between sampling grain and estimated niche width.	63
Fig. 2-12 Differences in generalism across grains by growth habit and native-status	64

Fig. 3-1. Hypothesized richness-generalism patterns in four pure models..	75
Fig. 3-2. Simulation model results for four pure scenarios.	83
Fig. 3-3. Generalism and richness patterns in plants of the Carolina Vegetation Survey.	85
Fig. 3-4. Generalism-richness relationships in CVS communities..	87
Fig. 3-5. Loess-smoothed relationship between plot richness and mean plot generalism in the six most common community types in CVS.	88
Table 3-5. Communities in CVS matching each of the proposed models	89
Figure 3-6. Richness and generalism relationship in two human systems	91

Acknowledgments

I must give thanks for the feedback provided me by my advisory committee, as well as Dr. Jack Weiss, regarding some statistical matters, and my colleague Nick Adams.

Preface

The niche concept has been with ecology for a long time (Grinnell, 1924), but its meaning has varied between fields and individual researchers; shifted over time; and has been debated, perhaps, *ad nauseum*. Some have abandoned the term as too amorphous and laden with baggage to be useful, while others have tried to reformulate it (e.g., Leibold, 1995). But not all ecological terms must be precise to be useful: “competition,” for instance, has had no less tangled a history, yet still refers to an idea that ecologists understand, at least in broad outline.

“Niche” may mean a species’ response to the environment: that is, its requirements. It may include the affect the species has on the environment; so that when all niches of a community are taken together, there is an endless dynamic feedback—which the first definition leaves to other concepts. It may be meant more broadly still, as a species’ total role within a community, encompassing all its interactions with all other species and the environment (see Vandermeer, 1972). And it may be focused as much on the ecosystem or surrounding community as on individual species: where a “niche” is an opening in some n-dimensional volume that is waiting to be filled by a species (Hutchinson, 1957).

All of these concepts are valid. We need not settle on any one of them, because “niche” need not be a piece of highly technical jargon; it may be merely a heuristic; a

symbol calling to an ecologist's mind such issues as where a species lives, and the interplay between the its traits, its environment, and other organisms. Such questions would seem to encompass much of ecology; but the focus, when we invoke the word "niche," is often something Hutchinsonian: a visual metaphor in which space is filled with shapes representing species—perhaps normal-shaped curves on a single line. And it is this metaphor which, however imperfect, guides much niche-based thinking. It engages our spatial processing: we intuit easily that shapes may not easily overlap because they have substance, there are differing arrangements and packing that may be more or less efficient. Without metaphors, scientists must merely stare at data or catalog first-hand experiences.

WHAT TO EXCLUDE FROM THE NICHE?

As just described, "niche" deals with a species and its community on the scale of abiotic and inter-specific interactions (but obviously has repercussions at much larger scales, say in distributions). But some categories of events are usually omitted, namely history. Ecologists like to focus on species-community interactions that are reproducible and a common factor wherever the species occurs. But identifying commonalities is always a matter of generalizing by omitting messy details, because all locations are unique, and all instances of a species occurrence different, dependent on innumerable factors. It is arbitrary what we consider worthy of categorizing and generalizing about and what we don't.

The difficulties of making such decisions can be illustrated by the rescue effect. Consider a site on the edge of a species' range. Asking whether the species can

survive there, we might consider major abiotic factors: climate, disturbance regime, etc. If these are only barely sufficient for survival, a single propagule may not be able to establish a self-sustaining population. It then becomes a requirement that a source population be nearby. The location of such a source may be seen merely as contingency. But what if the species inherently exists in meta-populations, rapidly colonizing disturbed sites? What if it is a generalist, and exists all over the globe? Different life history strategies make the existence of a source more or less likely.

There is, then, a continuum of factors affecting species' existence: some stable and easy to generalize about, like climate; others dynamic and harder to measure at a single point in time, like colonization history. One is not inherently more important than the other, and there is no reason for science to concern itself with one and not the other. But this is, in some ways, what we do by enclosing stable factors in the niche concept while excluding others as "merely" history. Of course, there are practical, rather than epistemological reasons to do things this way. (It may also be a human or Western bias to favor seemingly Platonic traits.)

An infinitely expansive niche concept would exclude nothing: the n-dimensional hyperspace would have axes for "distance to nearest source population," and "time since last disturbance," among many others. This is essentially "realized niche" in common parlance. It is all we can measure with observational data. We assume there is a correlation between the realized niche and any other we care to define, but experimental research on particular species and ecosystems must be used to confirm that.

If the expansive, realized niche concept is all we have though, we cannot single out factors controlling a species' distribution; it allows no such distinctions to be made. Thus, even an arbitrary definition of niche, which excludes some variables, has utility. A common one eliminates recent history, stochastic events, and geography; and it seems as good as any.

NICHE BREADTH

No matter the niche concept we use, it will undoubtedly be complex. For it to be useful as an ecological tool, it must be characterized simply, in a way that is universal. Perhaps the coarsest possible summary is niche size. A large niche features a greater variety of interactions—with the environment, predators, mutualists, and so on. But the precise meaning depends on what variables we have permitted in the niche concept. If we use the common definition of niche given above, and focus on the requirements of existence, niche size is the variety of circumstances a species can tolerate. And we call this generalism.

The following work attempts to show the utility of generalism as a multi-purpose tool for ecologists. First, it demonstrates a method for estimating niche width with large-scale abundance data. Second, it expands the idea of niche to describe species tolerance across scales. Third, it considers the use of species generalism to describe whole communities, and to infer the processes that maintain them.

WORKS CITED

Grinnell, J. (1924) Geography and Evolution. *Ecology*, **5**, 225–229.

Hutchinson, G. E. (1957) Concluding Remarks. *Cold Spring Harbor Symposia on Quantitative Biology*, **22**, 415–427.

Leibold, M. A. (1995) The Niche Concept Revisited: Mechanistic Models and Community Context. *Ecology*, **76**, 1371–1382.

Vandermeer, J. H. (1972) Niche Theory. *Annual Review of Ecology and Systematics*, **3**, 107–132.

1. Variations in Theta

Designing a refined niche width metric based on co-occurrence

INTRODUCTION

Ecologists have long sought to characterize species traits, including those that broadly describe ecological roles, such as generalism or niche breadth (defined formally by Hutchinson, 1959). Although some traits may only be ascertained through labor-intensive, per-species experiments (e.g. tree dispersal distances), the availability of large-scale datasets may allow others to be detected statistically for many species simultaneously. Estimation of niche breadth (habitat tolerance) has usually required extensive sampling of environmental variables throughout a species' range (e.g., Vetaas, 2002). However, Fridley et al. (2007; henceforth JF07) demonstrated that co-occurrence data alone might be suited to this purpose.

JF07's authors reasoned that a generalist species, which by definition can tolerate many conditions, should appear in more communities and thus co-occur with more species than would a specialist, assuming regular turnover across a heterogeneous landscape. Their "theta" metric counted the number of co-occurers for each focal species in a dataset (with corrections for site richness and non-random sampling).

Variations on richness correction techniques have been debated already, but I wondered whether several other small changes might improve the metric's ability to

estimate niche breadth. I framed these as options within five discrete data-processing steps, where all permutations create valid niche widths metrics. These were assessed using a spatially implicit simulation model, where true species widths could be known and compared to each metric's estimates. With this approach, I essentially asked three questions:

1. How is metric performance (in co-occurrence-based metrics generally) affected by different kinds of niche distribution, particularly as this controls saturation?
2. How do different metric variations compare? Is it helpful (2a) to consider abundance or combinations of species, (2b) to correct for richness by co-occurrence richness, (2c) omit some occurrences, or (2d) normalize species abundances? Finally, (2e) does the sampling regime matter, and when is subsampling helpful?
3. Under what real-world circumstances—that are knowable—should we use one metric over another? Does any single metric stand out as broadly useful in many systems?

METRICS

Design Choices

A. Further Use of Data

The JF07 theta utilizes co-occurrence data, but other information is commonly available from large datasets. I tested three algorithms for counting co-occurrences.

1. Binary Count

As in JF07, the total number of species that ever co-occur with the focal species is found.

2. Maximum Abundance

In the binary method above co-occurrences essentially act as milestones along a gradient which focal species can encounter. Their density controls the resolution of the measure: many species allow for fine discriminations of width. If there are few co-occurrences, the binary metric becomes very coarse. We can ameliorate this problem by utilizing co-occurrence abundances.

If a focal species's niche extends only so far as to partially intersect that of a co-occurrence, that co-occurrence's maximum abundance (within the focal range) will be low. If the focal species intersects its optimum though, the maximum abundance will be high. Therefore, we find the maximum abundance of each co-occurrence and sum them. This assumes that co-occurrence niches have about the same heights—or a correction can be made, which is discussed later.

3. Combinations

A generalist will tend not only to encounter more species, but more combinations of species. The difference between these statistics could emerge in two scenarios. First, niche might theoretically be discontinuous in niche space, i.e. possessing holes where some combinations of resource create intolerable conditions. Such a scenario is not straightforward to model, and may be rare in the real world.

Second, if niche overlap in a dataset is extremely high, all species will have the same number of co-occurers. This may easily occur if only a small portion of a key resource gradient is sampled. But there are still differences between generalists and specialists, even if all species co-occur: the generalists can occur with all other species, while some specialists may out-compete one another. To see this difference, we merely count the number of unique assemblages a focal species is part of.

Examples

Fig. 1-1 shows a possible set of niches and three different focal species widths that should be discriminated between by any metric. Each co-occurrence counting method (A, above) responds slightly differently.

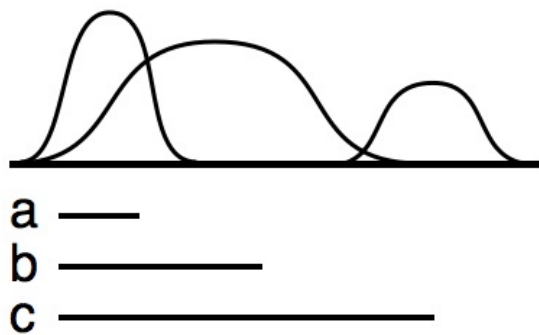


Fig. 1-1. Example niche curves and three possible focal species widths (a-c).

- Binary Count: Straightforwardly finds 2 spp. at (a), also 2 at (b) and 3 at (c).
- Combinatoric: Differentiates (a) from (b) because the middle co-occurrence is unlikely to appear by itself at (a). (c) adds 2 more combinations.
- Max-Abundance: (a) sees high abundance for the first spp. but only a low part of the second's curve, while at (b) the optima for both spp. are captured, raising the total. (c) Adds a further point for high abundance.

B. Richness Correction

One complication is that some species tend to occur at rich sites and pick up more co-occurrences. To calibrate counts to a standard scale and render them comparable, JF07 subtracted the mean richness of a focal species' sites from its total count. Zeleny (2008) pointed out that this makes sense only if sites are saturated, so that local richness is divorced from the size of the species pool; if this is not the case, dividing by mean richness is more effective. Either action turns the metric into a kind of beta-diversity, and many measures of beta diversity could be utilized (Manthey and Fridley, 2008).

Division of the co-occurrence count by mean richness corrects for local richness as long as sites are not saturated. A further assumption of this technique is that site richness does not vary considerably across the focal species' range. But generalist species may have large ranges, and occur in sparse and dense regional species pools.

To account for this possibility, one can instead find the mean richness for *each co-occurren* (only where it occurs with the focal species), and take the inverse to weight each co-occurren by its richness. With the binary count method above (A1), the count is multiplied by an average of these weights. The combinatoric and maximum-abundance methods (A2, A3) produce values for each co-occurren, so those are directly multiplied by the weights before the quantity is averaged.

I tested this method as well as simple division and subtraction of mean richness (“additive” and Whittaker’s Beta, respectively).

C. Data Subsetting

Species often fail to occur where they could based solely on abiotic conditions (which co-occurrens are a proxy for). Such “omissions” are more likely in sites that are marginal for a species—where its abundance will tend to be low. But problematically, these are exactly the sites that define the extremes of a species’ range and hence its generalism. However, if niche curves are symmetrical then measuring only the upper portions of the curve, where the species is abundant, will not change estimates of its niche width, relative to those of other species.

I tested subsetting of the data to exclude occurrences below a certain abundance; specifically below a selected *quantile* at that location, rather than a fixed threshold, as this would transfer poorly between datasets with different richnesses.

D. Normalization of Abundances

Subsetting data may have the side-effect of excluding inherently rare species that are never very abundant. The “maximum abundance” method for step A also tends to under-represent rare species. To compensate for this, we could normalize abundance data for each species, comparing its abundance at any site to its typical abundance across the dataset.

Depending on the method used in A, rare species may be more important than common species, so such a correction could be important. While a linear transformation could work, it neglects the central tendency of a species’ abundances, so I used cumulative density functions instead.

E. Sampling Regime

JF07 called for subsampling of the dataset, for instance 100 groups of 50 plots for CVS data, to reduce the effects of bias in the original site selection process. However, increasing sample number at the expense of size is not necessarily of benefit to all metrics; and some may be less susceptible to bias in the first place. Further, biased selection may have different effects depending on the species involved (e.g., whether niches are evenly distributed or clustered in niche space).

Performance Under Different Scenarios

Any metric should be demonstrated to function in a variety of scenarios. Zeleny (2008) showed that site saturation was critical for metric performance. Local conditions and

the density of niches in resource space will control saturation. Niche density will depend on the number of species captured in a study, the size and distribution of niche widths, and their arrangement. These properties also may have other effects on co-occurrence that have little to do with saturation. I tested all metrics in a variety of scenarios, with different niches.

If mean niche width is small, relative to the scale of the study, there will be little overlap between them and thus little co-occurrence to base niche-width estimates on. Conversely, a single study will rarely encompass the totality of variation present on the planet for any key environmental gradient. Niche widths may thus be larger than the section of niche-space in the sample, and thus be unmeasurable; and if most of the focal species have large niches, relative differences will be slight. The distribution of widths around a mean may also matter: generally most species are fairly rare while a few are common, and thus could be due to niche size. But if widths were drawn from a more uniform distribution—or appeared to be so at the scale of the study—they might have a very different pattern of co-occurrence, relevant to our purpose.

The positioning of niches through space should also affect co-occurrence patterns. Theoretically, if niche optima are equally spaced (because of niche differentiation or character displacement) it should make co-occurrence rates dependent on width and sampling alone, and thus make width easier to estimate. If the niches are instead randomly distributed, or clustered at the favorable portion of a gradient, detection should be harder.

METHODS

The metric-construction options described can be performed in any combination, resulting in a large (technically infinite) number of metrics. To evaluate them, I created a framework in R to compute the options of each choice in parallel and return width estimates for all species in a dataset. Abundance normalization was computed first, followed by subsetting, co-occurrence counting methods, and finally richness correction. When random subsampling of the data was performed to defeat bias in site selection, the results from all samples were averaged. Initial trials suggested that subsetting values above 0.6 were extremely harmful to metric performance so values of 0, 0.2, 0.4 and 0.6 were tested. In total 192 ($2 \times 4 \times 3 \times 4 \times 2$) metric variants were compared.

Evaluation by Simulation

The performance of these metrics was established by a simulation model implemented from scratch in R 2.8.3 and executed on UNC's Emerald Academic Computing Cluster. This model largely followed that used in JF07 but different in some details. Various parameters were varied to create 72 different scenarios, each of which was replicated 100 times. (For code, see Appendix C.)

50 species niche curves were first generated on a resource axis. Early trials showed that using multiple axes did not affect relative metric performance. JF07 demonstrated that niche shape was also largely irrelevant, so a normal shape was always used. In JF07 niche widths were drawn from a uniform distribution, ranging from 0.002 to 1 of the total gradient length. I tested both this and a gamma function in order to generate the familiar "hockey-stick" curve of species commonness; various mean values were used

(Table 1-1). Niche optima were variously spaced equally along the gradient, positioned randomly, or positioned randomly with a bias toward one end.

Plots were assigned resource levels chosen randomly with bias toward one end of the gradient, representing site selection bias (this was the same end that species optima could be biased toward). Each plot was then populated by drawing species from the pool. The probability of selection was equal to the height of a species' niche curve at the given resource level. 100 draws were performed at each plot. A species could be drawn multiple times, and this provided relative abundances.

Table 1-1. Simulation parameters were varied to create 72 distinct scenarios.

parameter	values
mean niche width	0.05, 0.2, 0.5, 1.0, 2.0
width distribution	gamma, uniform
niche position method	random uniform, equal spacing, biased
site bias	no, yes

Linear models were used to compare the generated niche widths to the estimates made by each metric; r^2 providing a simple performance score. (It was preferred to r , correlation, for being stricter and more readily interpretable.)

In JF07, species niches were not allowed to fall outside the measured gradient but were instead truncated. This is perhaps reasonable but assumes that the sampled gradient captures the entire range of variation. However, no real-world dataset is likely to be so

complete. I allowed niches to extend beyond the measured range, thus penalizing metrics for underestimating those widths.

RESULTS

The implementation here appears to adequately reproduce the simulation algorithms of JF07. In the scenario they employed (not part of results below) the estimated widths were correlated with true, generated, widths to a similar degree: at $r^2 = 93.1$ here, while they reported 92.5.

Performance and Niche Properties

To consider the effects of the simulation parameters, the mean performance of all 192 metrics was found in each of the 72 scenarios. First consider width generation. Peak performance was seen with mean widths around 0.2 of the sampled gradient (Fig. 1-2); this gradually declined as widths increased and sharply declined as they approached 0. For any mean, widths generated via a gamma function were easier to estimate, especially for widths smaller than 1.0.

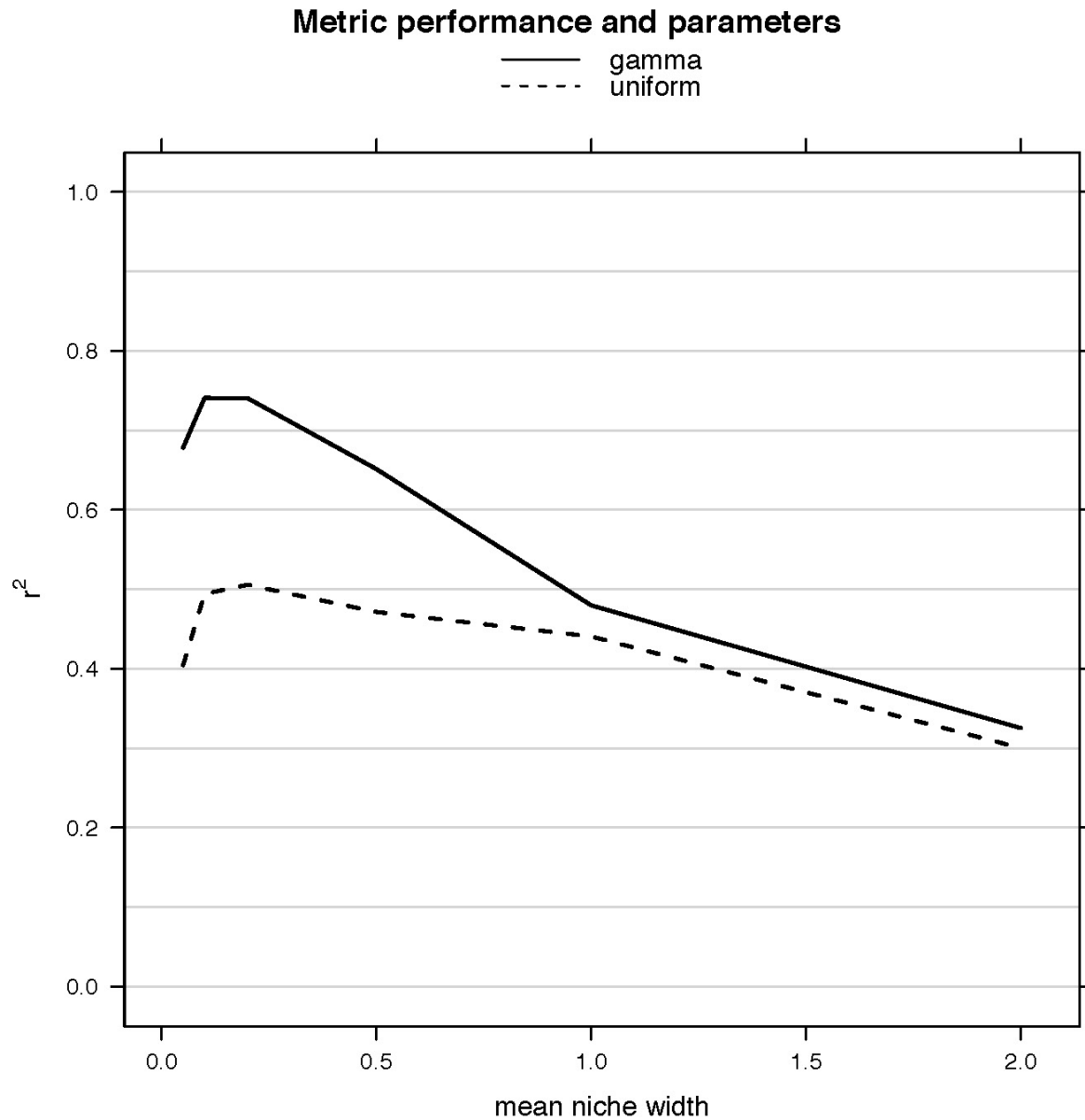


Fig. 1-2 Effect of niche width means and distribution on estimates (r^2).

Biased site selection led to estimates around 0.05 r^2 lower (Fig. 1-3). Niches that were evenly spaced produced the best estimates while biased gave the worst, though the differences were small.

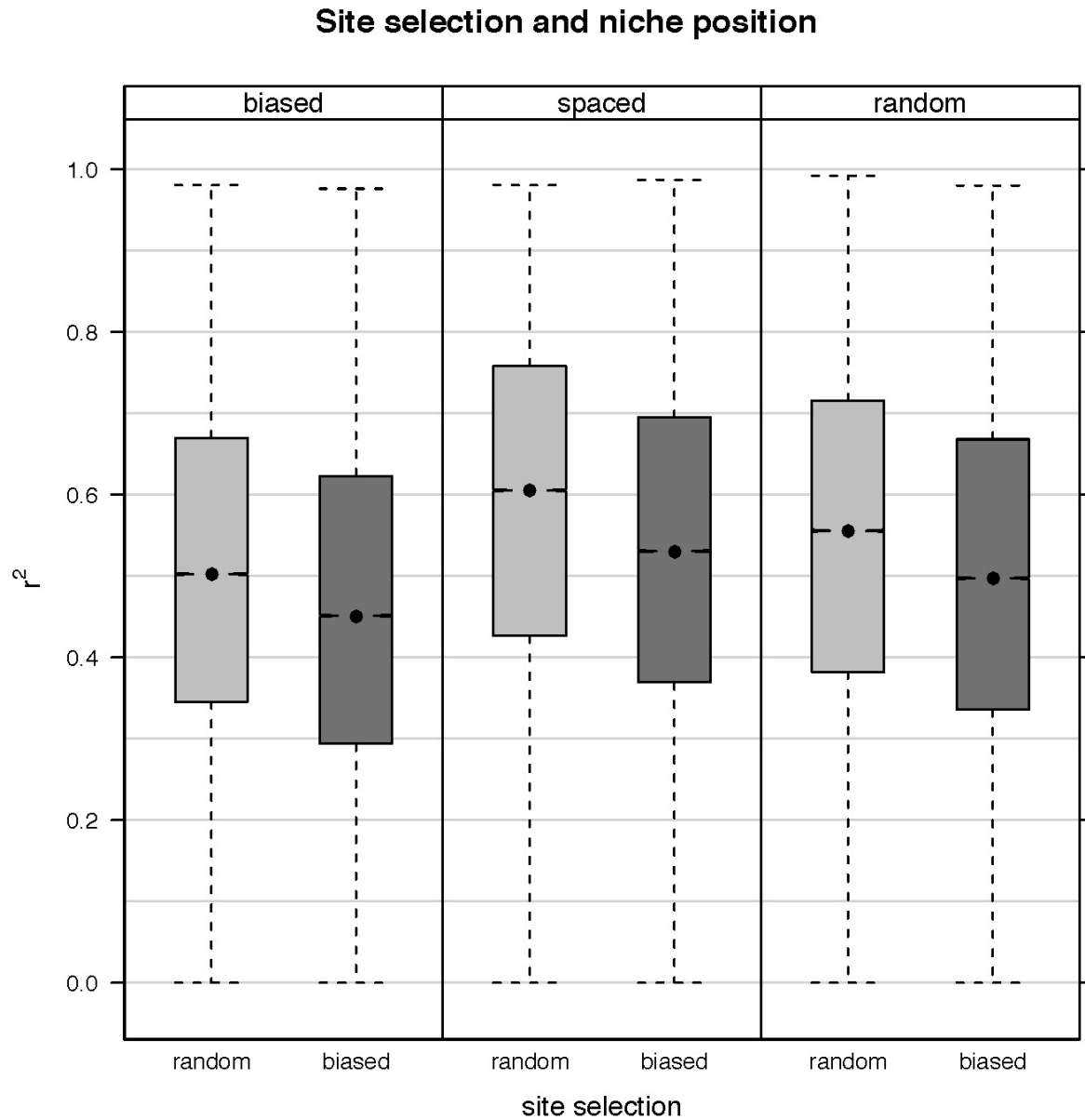


Fig. 1-3. Positioning of sites (abscissa) and niche optima (panels) on a gradient.

Metric Choices

ANOVA showed that the choice made for each metric-constructing option, as well as each varied parameter, had significant effects on r^2 , as did all 2- and 3-way interactions (see Appendix A). Not all interactions had large estimated effects, though.

Plot sub-sampling tends to harm performance. It is designed to ameliorate the effects of site selection bias, but with such bias, sub-sampling merely harms performance less; particularly if normalization is not employed, in which case it almost achieves parity (Fig. 1-4).

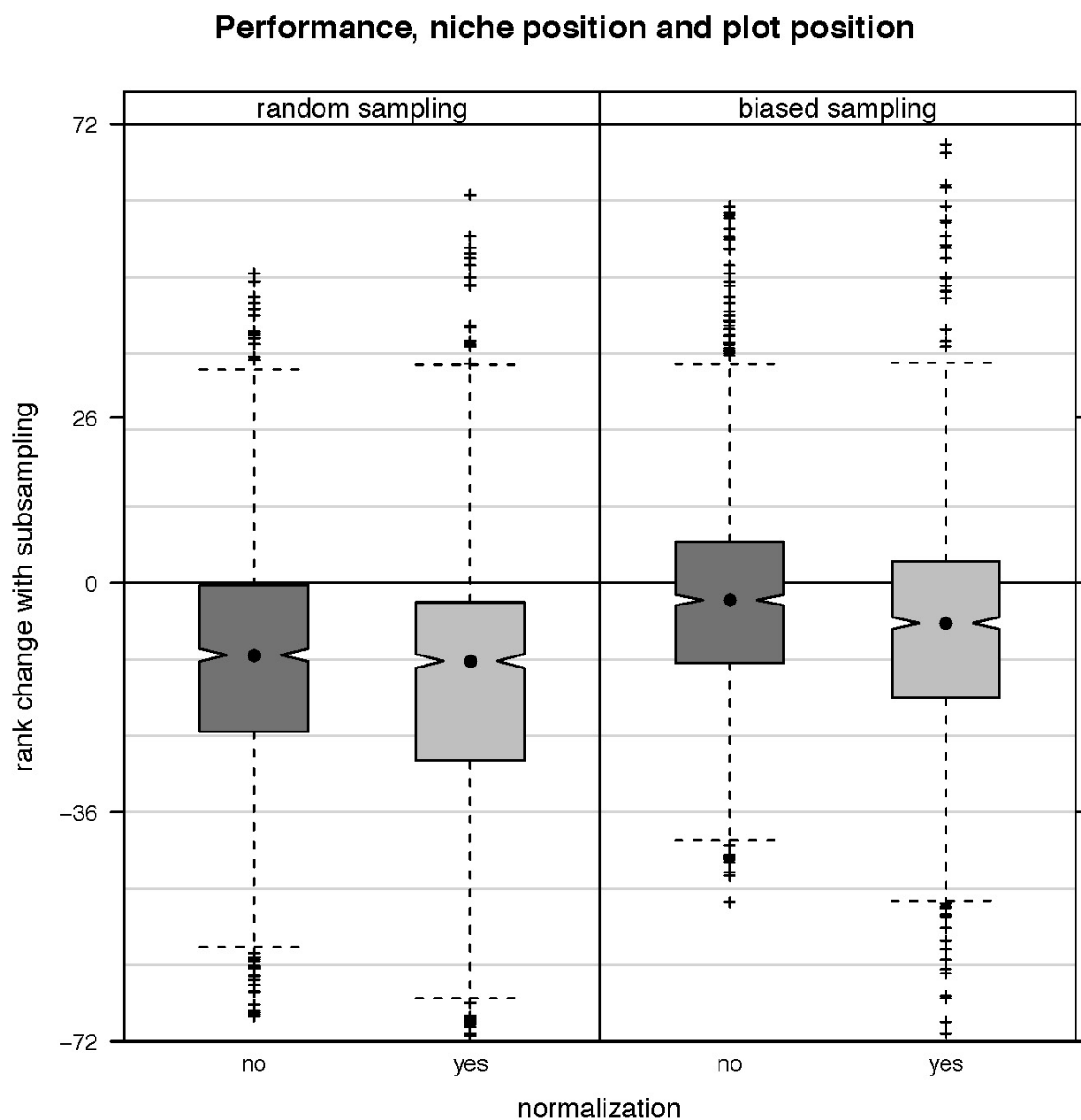


Fig. 1-4. Increase in ranked metric performance by sub-sampling (taking 10% of available plots 100 times and averaging the results); zero line shows no difference. Panels compare uniform and biased site sampling on the gradient. Light bars show the effect of abundance normalization.

In general, however, normalization tends to improve estimate quality. Consider its effects on the three co-occurrence counting methods, and subsetting. Ordinarily, subsetting severely decreases mean rank (Fig. 1-5). But when abundances are

normalized subsetting can be beneficial at a low level. Normalization also helps the combinatoric and maximum-value methods more generally; benefitting the latter enough to broadly win out over a simple count.

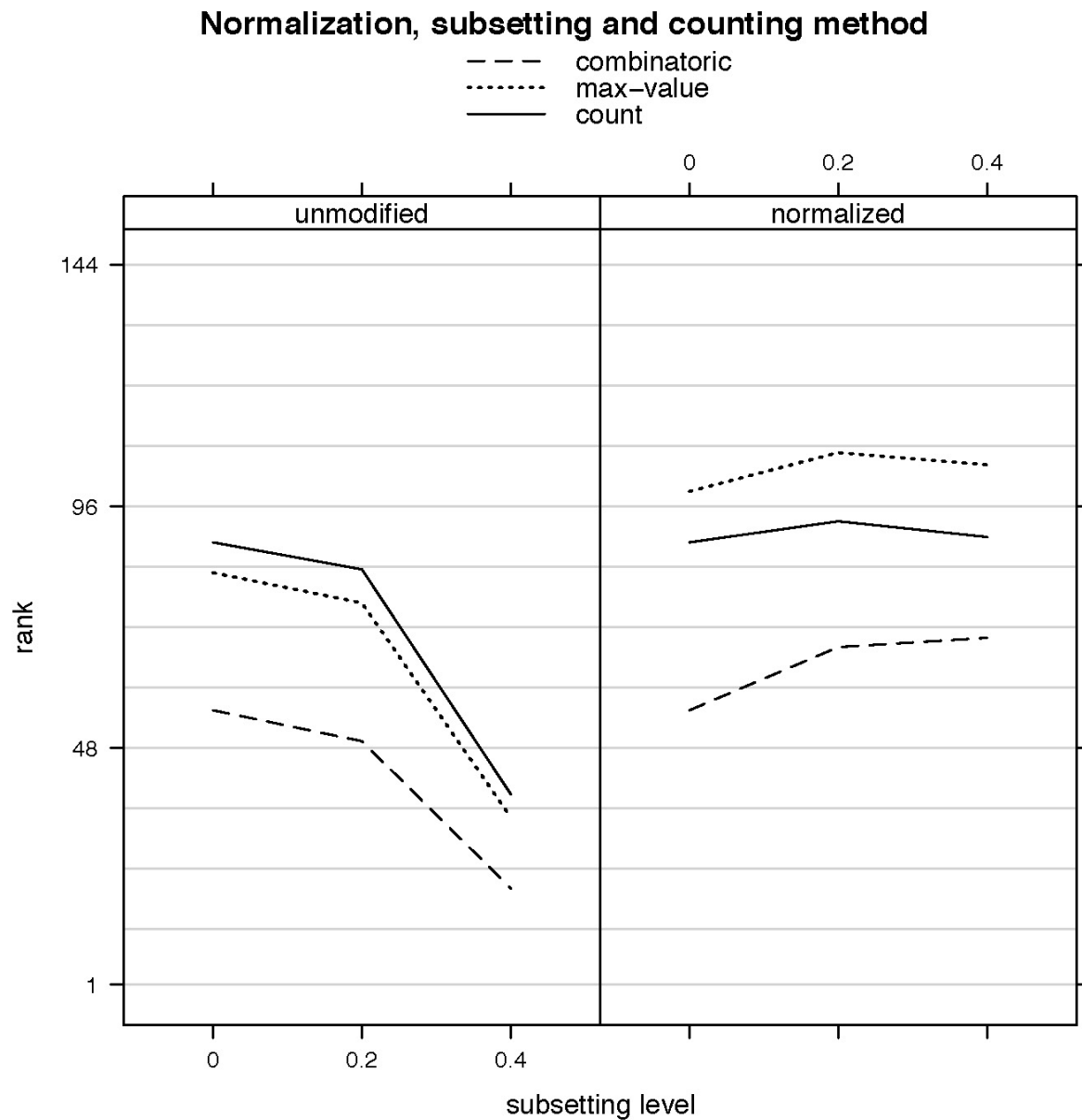


Fig. 1-5. Rank performance of three co-occurrence counting methods with and without normalization and under different levels of subsetting.

The method of counting co-occurrences also depends heavily on the distribution of niche widths in the simulation. Combinatoric methods do poorly overall but better under extreme circumstances: when niche widths are very small or very large (Fig. 1-6). Counting maximum values is usually superior to simple counting, but not when widths are uniformly distributed and typically larger than the sampled portion of the gradient.

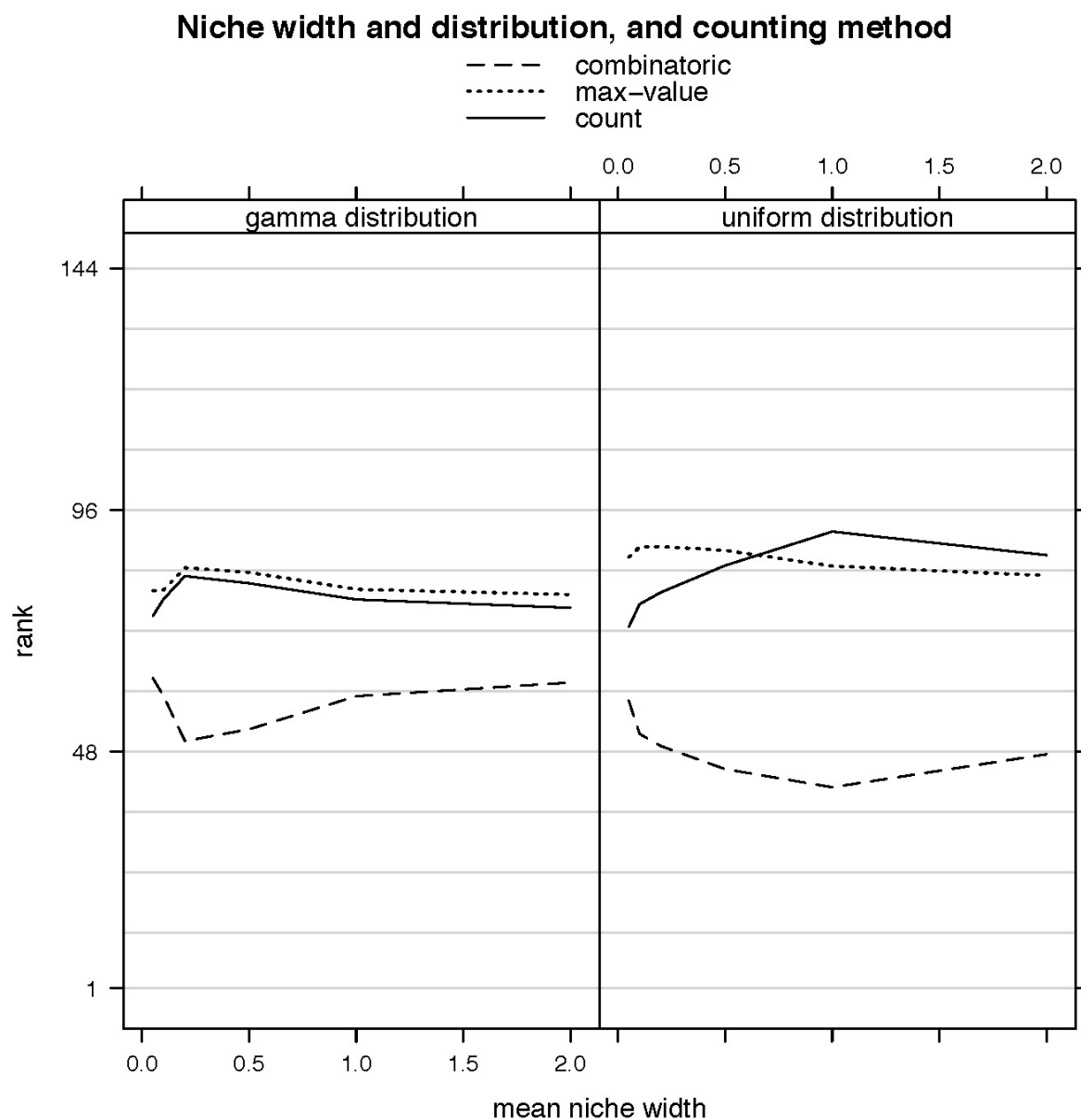


Fig. 1-6. Interaction of niche sizes with co-occurrence counting methods. Niche width means were varied and drawn from a uniform (right panel) or gamma distribution (left).

Niche width and arrangement contribute to mean plot saturation in a simulation, where saturation is defined as the number of species that could exist in a site divided by the actual richness. Generally, higher saturation decreases estimate quality (Fig. 1-7) but this depends greatly on the type of richness correction used. As others have noted

(Zeleny, 2008), subtraction of mean richness is less effective than division at low saturation but at higher levels (around 3.5 here) is more effective. This relationship holds for co-division, although it does poorly at extremely low saturations; it is generally better than basic division. A lack of richness correction is a large disadvantage at low to moderate saturation but preferable when saturation is high. On average, subtraction is never the best option.

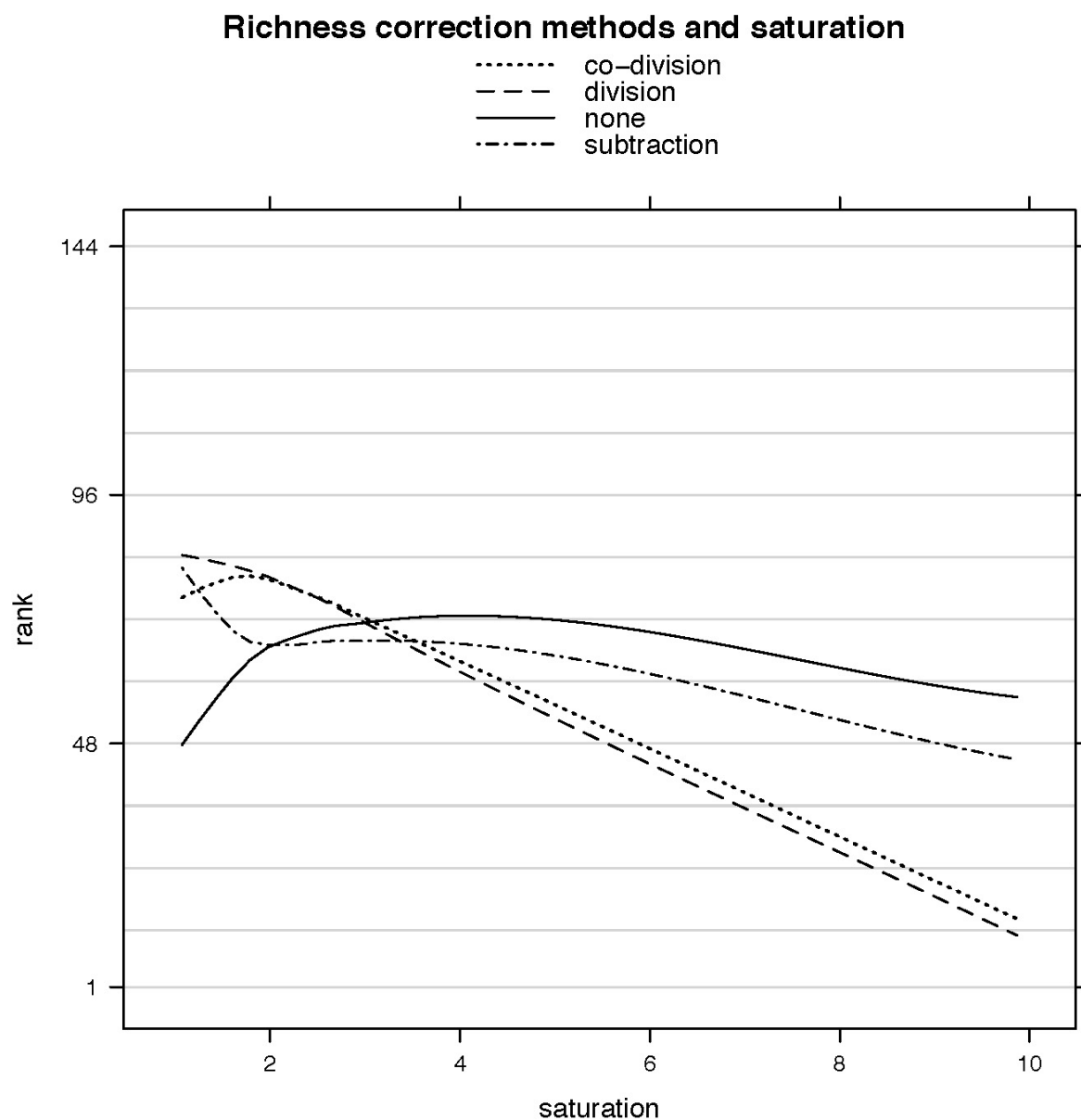


Fig. 1-7. Effect of plots saturation on metric rank for four richness correction methods.

Metric Selection

No single metric was consistently the top-performer in all scenarios. The one with the highest mean r^2 was, in my numbering scheme, #55n, using normalization, 0.2-level subsetting, maximum-value counting and division by mean richness, with no sub-

sampling. While mean r^2 is surely important, it does bias the selection toward the particular parameters used even though these may not represent what is common in the real world. Of equal importance may be reliability across scenarios: the metric with the highest minimum r^2 was #67n, identical to #55n except using 0.4 level subsetting. But by Tukey's Honest Significant Difference test, differences between the two metrics may be due to chance alone.

We may contrast the above metrics with three alternatives (Table 1-2). Niche width may naively be equated with species commonness. The quality of such estimates is often decent but depends greatly on the scenario and was observed dropping to 0. A straightforward count of co-occurrences with no manipulations or sub-sampling (#1n) similarly had very low minimum performance, though beat JF07 when mean niche width was high (Fig. 1-8). Both were consistently beaten by #55n in most scenarios; the exception being that JF07 had better minimum performance when niches were extremely small. JF07 and the alternative proposed by Zeleny (2008), using division-based richness correction (#3y), performed indistinguishably according to Tukey's HSD test.

Table 1-2. Performance selected metrics, as well as the pseudo-metric of species commonness, in all scenarios and replicates. The last column indicates statistically different groups.

metric	norm.	subset	co-occ.	rich	sub	mean	min	sig
commonness	–	–	–	–	no	0.52	0.00	a
1n	no	0	count	none	no	0.55	0.01	b
2y (JF07)	no	0	count	subtract	yes	0.58	0.02	c
3y	no	0	count	divide	yes	0.57	0.03	c
55n	yes	0.2	max-v	divide	no	0.66	0.11	d
67n	yes	0.4	max-v	divide	no	0.65	0.12	d

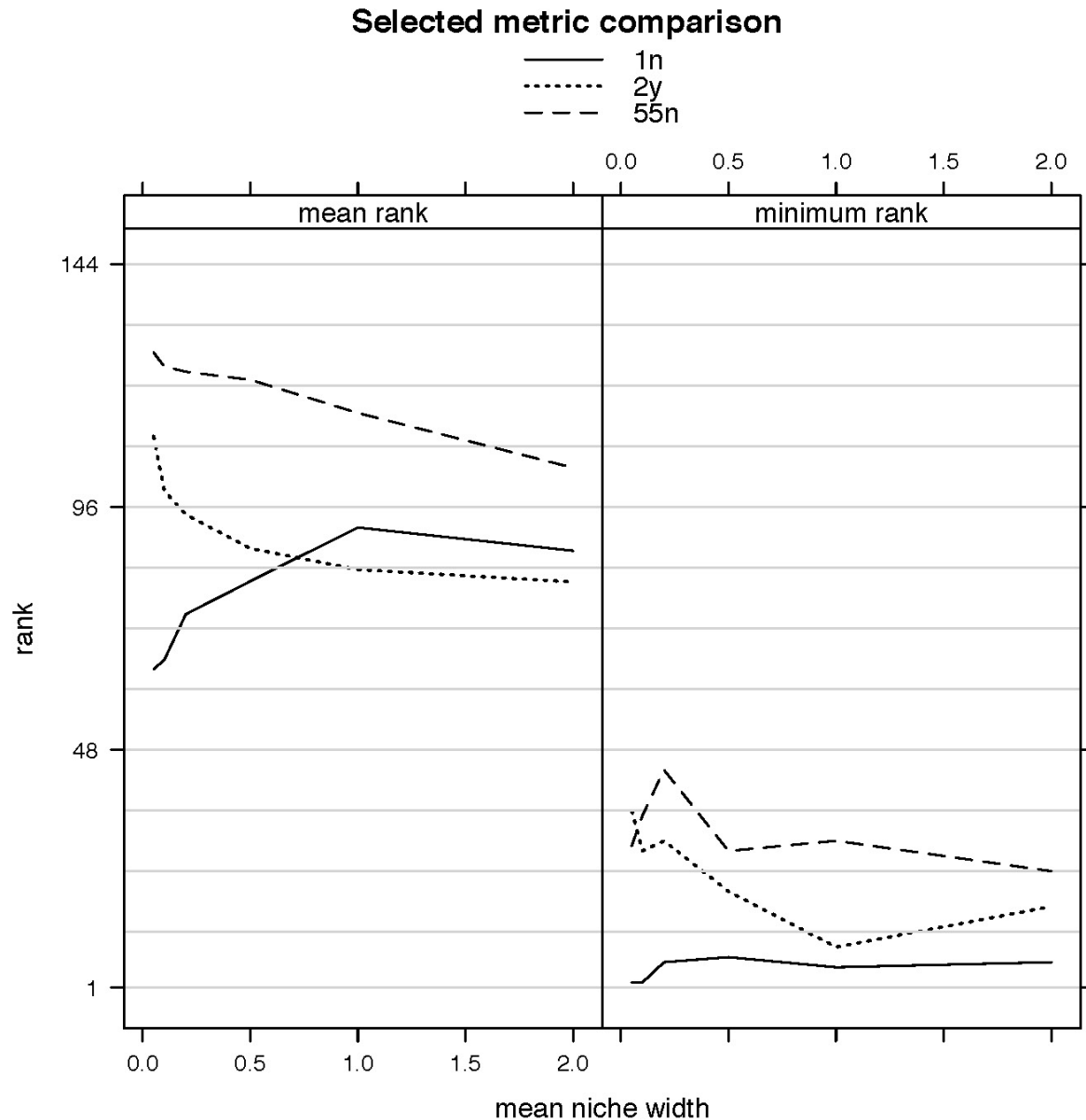


Fig. 1-8. Performance of selected metrics with various mean widths, with in-replicate rank used for clarity. Left and right panels show mean and minimum rank. Metric #2y was that proposed in JF07.

Metrics #1n, #2y and #55n were applied to woody plants in the Carolina Vegetation Survey database (603 species in 2804 plots). The estimates for each, as well as commonness, were scaled from 0 to 1 to allow comparison. Commonness was strongly

correlated with estimates ($r^2 < 2.2 \times 10^{-16}$, $p < 0.001$) but many species were less common than their niches would suggest (Fig. 1-9).

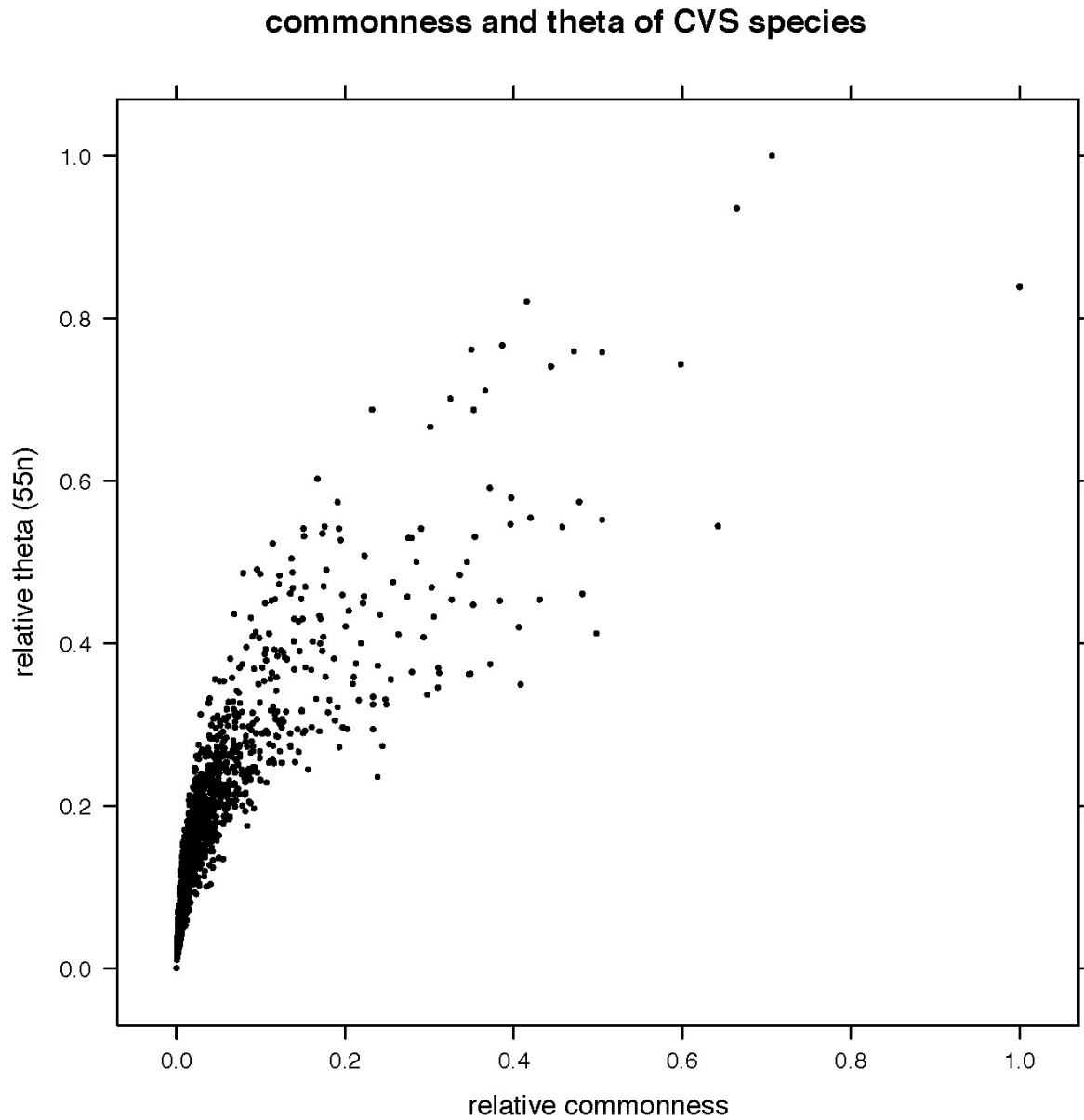


Fig. 1-9. Scaled commonness and niche width estimates from metric #55n of CVS woody plants.

Many species were awarded similar generalism scores by the selected metrics, but some showed large differences (Table 1-3). *Acer rubrum* (red maple) was the most common

species, occurring in 67% of plots; #2y ranked it the most general species but #55n did not, giving it a score of 0.84 instead. The most general species according to #55n was *Smilax rotundifolia* (common greenbriar), occurring in 47% of plots; but rated only 0.77 by #2y. Consider also two less common species. Farkleberry (*Vaccinium arboreum*) occurs throughout the Southeastern US and in 7% of CVS plots: its width was estimated to be 0.19 by metric #2y, 0.45 by #55n. *Fraxinus profunda* (pumpkin ash) occurs in near 1% of plots and is rated 0.00 and 0.09 by #2y and #55n.

Table. 1-3. Woody plants occurring in at least 0.5% of plots in the CVS database (N=287) with commonness, federal native status and growth habit (USDA) and generalism estimates by three metrics.

name	common	#1n	#2y	#55n	growth habit	native status
<i>Abies fraseri</i>	2.7%	0.12	0.02	0.14	Tree	N
<i>Acer floridanum</i>	3.1%	0.29	0.06	0.18	Tree	N
<i>Acer leucoderme</i>	0.7%	0.15	0.01	0.09	Tree, Shrub	N
<i>Acer negundo</i>	6.0%	0.34	0.11	0.24	Tree	N
<i>Acer pensylvanicum</i>	27.4%	0.41	0.36	0.35	Tree, Shrub	N
<i>Acer rubrum</i>	67.2%	1.00	1.00	0.84	Tree	N
<i>Acer saccharinum</i>	0.9%	0.09	0.01	0.08	Tree	N
<i>Acer saccharum</i>	16.4%	0.33	0.27	0.27	Tree, Shrub	N
<i>Acer spicatum</i>	4.7%	0.21	0.07	0.20	Tree, Shrub	N
<i>Aesculus flava</i>	15.7%	0.35	0.27	0.29	Tree, Shrub	N
<i>Aesculus pavia</i>	1.1%	0.21	0.03	0.10	Tree, Shrub	I
<i>Aesculus sylvatica</i>	2.2%	0.25	0.04	0.14	Tree, Shrub	N
<i>Ailanthus altissima</i>	0.6%	0.22	0.01	0.12	Tree	I
<i>Albizia julibrissin</i>	0.6%	0.27	0.02	0.13	Tree, Shrub	I
<i>Alnus serrulata</i>	1.7%	0.38	0.05	0.24	Tree, Shrub	N
<i>Amelanchier arborea</i>	2.1%	0.25	0.04	0.22	Tree, Shrub	N
<i>Amelanchier canadensis</i>	1.5%	0.27	0.03	0.25	Tree, Shrub	N
<i>Amelanchier laevis</i>	14.9%	0.44	0.27	0.46	Tree, Shrub	N
<i>Amelanchier obovalis</i>	0.7%	0.18	0.01	0.11	Shrub	N
<i>Amorpha herbacea</i>	0.9%	0.14	0.01	0.13	Shrub	N
<i>Ampelopsis arborea</i>	4.5%	0.35	0.09	0.33	Vine, Shrub	N
<i>Amphicarpaea bracteata</i>	11.4%	0.58	0.31	0.43	Vine, Forb/herb	N
<i>Apios americana</i>	0.6%	0.34	0.03	0.17	Vine, Forb/herb	N
<i>Aralia nudicaulis</i>	2.5%	0.19	0.05	0.19	Subshrub, Forb/herb	N
<i>Aralia racemosa</i>	1.9%	0.20	0.04	0.13	Subshrub, Forb/herb	N

<i>Aralia spinosa</i>	2.0%	0.30	0.05	0.23	Tree, Shrub	N
<i>Aronia arbutifolia</i>	7.6%	0.47	0.17	0.36	Shrub	N
<i>Aronia melanocarpa</i>	0.9%	0.13	0.01	0.13	Shrub	N
<i>Asimina angustifolia</i>	0.7%	0.12	0.02	0.09	Shrub	N
<i>Asimina parviflora</i>	0.8%	0.14	0.01	0.11	Tree, Shrub	N
<i>Asimina triloba</i>	6.3%	0.39	0.14	0.29	Tree, Shrub	N
<i>Baccharis halimifolia</i>	1.5%	0.32	0.05	0.24	Tree, Shrub	N
<i>Berchemia scandens</i>	6.6%	0.43	0.13	0.41	Vine	N
<i>Betula alleghaniensis</i>	16.7%	0.30	0.24	0.33	Tree	N
<i>Betula lenta</i>	20.8%	0.42	0.32	0.35	Tree	N
<i>Betula nigra</i>	1.5%	0.24	0.03	0.13	Tree	N
<i>Bigelowia nudata</i>	3.7%	0.28	0.09	0.18	Subshrub, Forb/herb	N
<i>Bignonia capreolata</i>	11.8%	0.57	0.27	0.54	Vine	N
<i>Callicarpa americana</i>	3.9%	0.41	0.09	0.31	Shrub	N
<i>Calycanthus floridus</i>	8.5%	0.34	0.15	0.30	Shrub	N
<i>Campsis radicans</i>	11.2%	0.61	0.25	0.60	Vine	N
<i>Carpinus caroliniana</i>	13.2%	0.64	0.32	0.46	Tree, Shrub	N
<i>Carya alba</i>	18.7%	0.69	0.44	0.53	Tree	N
<i>Carya aquatica</i>	1.9%	0.17	0.03	0.15	Tree	N
<i>Carya carolinae-septentrionalis</i>	0.6%	0.14	0.01	0.07	Tree	N
<i>Carya cordiformis</i>	13.7%	0.56	0.33	0.44	Tree	N
<i>Carya glabra</i>	28.2%	0.70	0.52	0.55	Tree	N
<i>Carya myristiciformis</i>	0.5%	0.14	0.00	0.06	Tree	N
<i>Carya ovalis</i>	1.6%	0.24	0.05	0.16	Tree	N
<i>Carya ovata</i>	4.2%	0.46	0.11	0.26	Tree	N
<i>Carya pallida</i>	3.8%	0.38	0.10	0.28	Tree	N
<i>Castanea dentata</i>	25.0%	0.37	0.34	0.37	Tree	N
<i>Castanea pumila</i>	4.5%	0.35	0.07	0.28	Tree, Shrub	N
<i>Ceanothus americanus</i>	1.9%	0.36	0.05	0.21	Subshrub, Shrub	N
<i>Celtis laevigata</i>	6.6%	0.34	0.13	0.27	Tree, Shrub	N
<i>Centrosema virginianum</i>	1.4%	0.30	0.03	0.18	Vine, Forb/herb	I
<i>Cephalanthus occidentalis</i>	1.5%	0.22	0.03	0.25	Tree, Shrub	N
<i>Cercis canadensis</i>	4.1%	0.46	0.10	0.31	Tree, Shrub	N
<i>Chamaecrista nictitans</i>	2.5%	0.40	0.06	0.20	Subshrub, Forb/herb	I
<i>Chamaedaphne calyculata</i>	0.5%	0.04	0.00	0.05	Shrub	N
<i>Chimaphila maculata</i>	25.0%	0.55	0.38	0.59	Subshrub	N
<i>Chionanthus virginicus</i>	3.1%	0.46	0.10	0.30	Tree, Shrub	N
<i>Cladrastis kentukea</i>	0.5%	0.08	0.00	0.05	Tree	I
<i>Clematis crispa</i>	0.6%	0.19	0.02	0.09	Vine	N
<i>Clematis viorna</i>	1.5%	0.37	0.06	0.21	Vine	I
<i>Clematis virginiana</i>	3.2%	0.38	0.08	0.24	Vine	N
<i>Clethra acuminata</i>	4.5%	0.23	0.06	0.27	Tree, Shrub	N
<i>Clethra alnifolia</i>	7.5%	0.43	0.20	0.32	Shrub	N

<i>Clitoria mariana</i>	5.0%	0.54	0.14	0.34	Vine, Forb/herb	N
<i>Cocculus carolinus</i>	1.2%	0.31	0.05	0.17	Vine	N
<i>Cornus amomum</i>	0.6%	0.28	0.21	0.29	Shrub	N
<i>Cornus asperifolia</i>	1.3%	0.21	0.02	0.10	Shrub	N
<i>Cornus florida</i>	32.1%	0.74	0.04	0.11	Tree, Shrub	N
<i>Cornus foemina</i>	0.3%	0.10	0.59	0.57	Tree, Shrub	N
<i>Cornus stricta</i>	0.9%	0.19	0.01	0.13	Tree, Shrub	N
<i>Corylus americana</i>	1.0%	0.27	0.03	0.13	Shrub	N
<i>Corylus cornuta</i>	2.0%	0.25	0.06	0.18	Tree, Shrub	N
<i>Crataegus</i>						
<i>macrocarpa</i>	3.2%	0.38	0.09	0.24	Tree, Shrub	N
<i>Crataegus marshallii</i>	1.4%	0.23	0.04	0.17	Tree, Shrub	N
<i>Crataegus</i>						
<i>phaenopyrum</i>	0.5%	0.08	0.00	0.08	Tree, Shrub	I
<i>Crataegus spathulata</i>	0.7%	0.16	0.02	0.12	Tree, Shrub	N
<i>Crataegus uniflora</i>	1.7%	0.38	0.06	0.23	Tree, Shrub	N
<i>Crataegus viridis</i>	0.8%	0.16	0.01	0.13	Tree, Shrub	N
<i>Cuscuta rostrata</i>	0.7%	0.10	0.01	0.13	Vine, Forb/herb	N
<i>Cyrilla racemiflora</i>	3.3%	0.31	0.07	0.29	Tree, Shrub	N
<i>Decumaria barbara</i>	2.6%	0.30	0.04	0.23	Vine	N
<i>Desmodium nuttallii</i>	1.7%	0.31	0.04	0.18	Subshrub, Forb/herb	N
<i>Desmodium</i>						
<i>viridiflorum</i>	0.8%	0.28	0.01	0.15	Subshrub, Forb/herb	N
<i>Diervilla sessilifolia</i>	0.7%	0.10	0.01	0.11	Shrub	N
<i>Diodia virginiana</i>	1.1%	0.26	0.04	0.15	Subshrub, Forb/herb	N
<i>Dioscorea villosa</i>	1.3%	0.28	0.03	0.20	Vine, Forb/herb	N
<i>Diospyros virginiana</i>	23.5%	0.83	0.59	0.76	Tree	N
<i>Elaeagnus umbellata</i>	0.9%	0.22	0.02	0.11	Shrub	I
<i>Epigaea repens</i>	9.4%	0.32	0.14	0.37	Subshrub, Shrub	N
<i>Erythrina herbacea</i>	0.6%	0.16	0.02	0.11	Tree, Subshrub, Shrub	N
<i>Eubotrys racemosa</i>	2.9%	0.36	0.07	0.28	Shrub	N
<i>Eubotrys recurva</i>	3.6%	0.18	0.05	0.18	Shrub	N
<i>Euonymus</i>						
<i>americanus</i>	15.0%	0.61	0.35	0.51	Forb/herb, Subshrub	N
<i>Euonymus obovatus</i>	2.2%	0.15	0.03	0.12	Shrub, Subshrub, Vine	N
<i>Euphorbia curtisii</i>	2.9%	0.26	0.06	0.21	Shrub	N
<i>Fagus grandifolia</i>	22.6%	0.55	0.42	0.48	Tree	N
<i>Fraxinus americana</i>	23.8%	0.65	0.49	0.53	Tree	I
<i>Fraxinus caroliniana</i>	3.6%	0.24	0.05	0.26	Tree, Shrub	N
<i>Fraxinus</i>						
<i>pennsylvanica</i>	8.5%	0.46	0.16	0.39	Tree	N
<i>Fraxinus profunda</i>	0.6%	0.08	0.00	0.09	Tree	N
<i>Galactia erecta</i>	2.7%	0.26	0.06	0.18	Forb/herb, Vine	N
					Subshrub, Shrub,	
<i>Galax urceolata</i>	18.8%	0.32	0.20	0.37	Forb/herb	N
<i>Galium aparine</i>	1.9%	0.28	0.03	0.20	Vine, Forb/herb	I
<i>Galium circaezans</i>	8.8%	0.55	0.24	0.38	Subshrub, Forb/herb	N
<i>Galium triflorum</i>	10.8%	0.53	0.29	0.40	Forb/herb, Vine	I
<i>Gaultheria</i>						
<i>procumbens</i>	2.8%	0.18	0.03	0.19	Subshrub, Shrub	N

<i>Gaylussacia baccata</i>	3.9%	0.22	0.05	0.20	Shrub	N
<i>Gaylussacia dumosa</i>	14.1%	0.42	0.30	0.35	Subshrub, Shrub	N
<i>Gaylussacia frondosa</i>	10.7%	0.48	0.26	0.37	Shrub	N
<i>Gaylussacia ursina</i>	16.0%	0.25	0.17	0.24	Shrub	N
<i>Gelsemium</i> <i>sempervirens</i>	9.2%	0.57	0.23	0.50	Vine, Shrub	N
<i>Gillenia trifoliata</i>	6.1%	0.28	0.12	0.25	Forb/herb, Subshrub	N
<i>Gleditsia triacanthos</i>	0.5%	0.20	0.01	0.10	Tree, Shrub	N
<i>Gonolobus suberosus</i>	1.2%	0.18	0.03	0.15	Vine, Forb/herb	N
<i>Gordonia lasianthus</i>	1.4%	0.12	0.01	0.13	Tree, Shrub	N
<i>Halesia tetraptera</i>	20.9%	0.42	0.32	0.36	Tree	N
<i>Hamamelis virginiana</i>	18.4%	0.53	0.31	0.46	Tree, Shrub	N
<i>Huperzia lucidula</i>	5.9%	0.27	0.10	0.27	Subshrub, Forb/herb	N
<i>Hydrangea</i> <i>arborescens</i>	3.3%	0.26	0.07	0.20	Shrub	I
<i>Hydrangea radiata</i>	1.3%	0.18	0.02	0.12	Shrub	N
					Subshrub, Shrub,	
<i>Hypericum cistifolium</i>	0.9%	0.19	0.00	0.13	Forb/herb	N
<i>Hypericum crux-</i> <i>andreae</i>	6.2%	0.38	0.17	0.25	Subshrub, Shrub	N
<i>Hypericum</i> <i>fasciculatum</i>	0.7%	0.15	0.00	0.13	Shrub	N
<i>Hypericum</i> <i>hypericoides</i>	10.3%	0.65	0.29	0.47	Subshrub, Shrub	N
<i>Hypericum nudiflorum</i>	0.7%	0.07	0.00	0.07	Shrub, Subshrub	N
<i>Hypericum prolificum</i>	0.6%	0.23	0.01	0.11	Shrub, Subshrub	N
<i>Hypericum</i> <i>suffruticosum</i>	0.9%	0.17	0.02	0.11	Shrub, Subshrub	N
<i>Ilex ambigua</i>	1.1%	0.24	0.03	0.20	Tree, Shrub	N
<i>Ilex coriacea</i>	3.4%	0.24	0.05	0.25	Tree, Shrub	N
<i>Ilex decidua</i>	8.0%	0.39	0.16	0.34	Tree, Shrub	N
<i>Ilex glabra</i>	12.1%	0.46	0.30	0.32	Shrub	N
<i>Ilex montana</i>	16.0%	0.34	0.24	0.37	Tree, Shrub	N
<i>Ilex myrtifolia</i>	1.4%	0.19	0.03	0.17	Tree, Shrub	N
<i>Ilex opaca</i>	24.6%	0.76	0.49	0.71	Tree, Shrub	N
<i>Ilex verticillata</i>	1.3%	0.26	0.02	0.20	Tree, Shrub	N
<i>Ilex vomitoria</i>	3.0%	0.30	0.05	0.20	Tree, Shrub	N
<i>Ipomoea pandurata</i>	0.9%	0.27	0.02	0.17	Vine, Forb/herb	N
<i>Itea virginica</i>	3.0%	0.27	0.04	0.26	Shrub	N
<i>Juglans cinerea</i>	1.1%	0.27	0.03	0.15	Tree	N
<i>Juglans nigra</i>	3.9%	0.46	0.13	0.25	Tree	N
<i>Juniperus virginiana</i>	7.1%	0.54	0.18	0.39	Tree	N
<i>Kalmia carolina</i>	0.8%	0.11	0.01	0.09	Shrub	N
<i>Kalmia cuneata</i>	0.2%	0.02	0.01	0.11	Shrub	N
<i>Lespedeza bicolor</i>	0.5%	0.13	0.03	0.07	Subshrub, Forb/herb	I
<i>Lespedeza cuneata</i>	0.5%	0.28	0.03	0.14	Subshrub, Forb/herb	I
<i>Leucothoe</i> <i>fontanesiana</i>	4.6%	0.27	0.07	0.23	Shrub	N
<i>Licania michauxii</i>	0.7%	0.12	0.01	0.10	Subshrub, Shrub	N
<i>Ligustrum sinense</i>	4.6%	0.42	0.11	0.32	Tree, Shrub	I

<i>Lindera benzoin</i>	8.0%	0.50	0.22	0.36	Tree, Shrub	N
<i>Liquidambar styraciflua</i>	23.7%	0.82	0.60	0.69	Tree	N
<i>Liriodendron tulipifera</i>	33.9%	0.73	0.55	0.55	Tree	N
<i>Lonicera japonica</i>	7.6%	0.62	0.20	0.45	Vine	I
<i>Lonicera sempervirens</i>	3.5%	0.42	0.08	0.26	Vine	I
<i>Lycopodiella alopecuroides</i>	2.0%	0.26	0.05	0.16	Subshrub, Forb/herb	N
<i>Lycopodiella appressa</i>	0.7%	0.16	0.02	0.08	Subshrub, Forb/herb	N
<i>Lyonia ligustrina</i>	9.2%	0.51	0.25	0.49	Shrub	N
<i>Lyonia lucida</i>	4.6%	0.37	0.10	0.31	Shrub	N
<i>Lyonia mariana</i>	7.4%	0.34	0.17	0.28	Shrub	N
<i>Magnolia acuminata</i>	14.5%	0.39	0.26	0.33	Tree	N
<i>Magnolia fraseri</i>	23.4%	0.35	0.28	0.36	Tree	N
<i>Magnolia grandiflora</i>	1.1%	0.17	0.02	0.12	Tree	I
<i>Magnolia tripetala</i>	0.7%	0.13	0.01	0.09	Tree	N
<i>Magnolia virginiana</i>	10.1%	0.47	0.23	0.43	Tree, Shrub	N
<i>Malus angustifolia</i>	0.5%	0.21	0.01	0.09	Tree, Shrub	N
<i>Menispermum canadense</i>	1.7%	0.27	0.03	0.20	Vine	N
<i>Menziesia pilosa</i>	1.5%	0.14	0.02	0.16	Shrub	N
<i>Mikania scandens</i>	2.8%	0.32	0.06	0.30	Vine, Forb/herb	I
<i>Mimosa microphylla</i>	2.6%	0.36	0.05	0.20	Vine, Forb/herb	N
<i>Mitchella repens</i>	21.8%	0.68	0.43	0.70	Subshrub, Forb/herb	N
<i>Morella caroliniensis</i>	3.5%	0.34	0.10	0.23	Tree, Shrub	N
<i>Morella cerifera</i>	5.9%	0.49	0.14	0.43	Tree, Subshrub, Shrub	I
<i>Morus rubra</i>	6.2%	0.50	0.15	0.37	Tree	N
<i>Nyssa biflora</i>	6.1%	0.40	0.03	0.21	Tree	N
<i>Nyssa sylvatica</i>	40.2%	0.82	0.72	0.74	Tree	N
<i>Oenothera humifusa</i>	0.6%	0.06	0.00	0.16	Subshrub, Forb/herb	N
<i>Opuntia humifusa</i>	2.0%	0.28	0.03	0.23	Shrub	N
<i>Osmanthus americanus</i>	1.9%	0.18	0.02	0.19	Tree, Shrub	N
<i>Ostrya virginiana</i>	6.0%	0.42	0.18	0.28	Tree, Shrub	N
<i>Oxydendrum arboreum</i>	32.3%	0.57	0.40	0.46	Tree, Shrub	N
<i>Parthenocissus quinquefolia</i>	31.7%	0.86	0.68	0.76	Vine	N
<i>Passiflora lutea</i>	4.4%	0.52	0.13	0.36	Vine, Forb/herb	N
<i>Pedicularis canadensis</i>	3.2%	0.29	0.08	0.21	Subshrub, Forb/herb	N
<i>Persea borbonia</i>	2.3%	0.19	0.03	0.15	Tree, Shrub	N
<i>Persea palustris</i>	10.2%	0.53	0.24	0.53	Tree, Shrub	N
<i>Persicaria sagittata</i>	0.5%	0.19	0.01	0.12	Vine, Forb/herb	N
<i>Phemeranthus teretifolius</i>	0.7%	0.19	0.02	0.11	Forb/herb, Subshrub, Subshrub, Shrub,	N
<i>Phlox nivalis</i>	1.1%	0.22	0.05	0.14	Forb/herb	N
<i>Picea rubens</i>	5.0%	0.19	0.06	0.21	Tree	N
<i>Pinus echinata</i>	3.6%	0.39	0.07	0.25	Tree	N
<i>Pinus elliotii</i>	1.4%	0.24	0.04	0.18	Tree	N

<i>Pinus glabra</i>	0.6%	0.14	0.02	0.09	Tree	N
<i>Pinus palustris</i>	17.1%	0.46	0.36	0.36	Tree	N
<i>Pinus pungens</i>	2.9%	0.18	0.03	0.13	Tree	N
<i>Pinus rigida</i>	8.2%	0.33	0.10	0.27	Tree	N
<i>Pinus serotina</i>	7.7%	0.42	0.18	0.32	Tree	N
<i>Pinus strobus</i>	20.0%	0.42	0.25	0.34	Tree	N
<i>Pinus taeda</i>	12.9%	0.63	0.33	0.54	Tree	N
<i>Pinus virginiana</i>	7.0%	0.41	0.13	0.35	Tree	I
<i>Planera aquatica</i>	1.1%	0.08	0.01	0.13	Tree	N
<i>Platanus occidentalis</i>	4.5%	0.36	0.09	0.28	Tree	N
<i>Polygonella polygama</i>	1.0%	0.11	0.01	0.14	Subshrub	N
<i>Populus deltoides</i>	0.7%	0.12	0.01	0.09	Tree	N
<i>Populus heterophylla</i>	2.1%	0.14	0.02	0.16	Tree	N
<i>Prunus caroliniana</i>	1.7%	0.21	0.02	0.17	Tree, Shrub	N
<i>Prunus pensylvanica</i>	4.8%	0.25	0.08	0.26	Tree, Shrub	N
<i>Prunus serotina</i>	29.9%	0.84	0.66	0.74	Tree, Shrub	N
<i>Prunus virginiana</i>	0.5%	0.12	0.00	0.06	Tree, Shrub	N
<i>Pterocaulon</i>						
<i>pycnostachyum</i>	4.6%	0.30	0.11	0.21	Forb/herb, Subshrub	N
<i>Pyrularia pubera</i>	12.2%	0.34	0.21	0.33	Shrub	N
<i>Pyxidanthra</i>						
<i>barbulata</i>	1.1%	0.10	0.01	0.10	Subshrub, Forb/herb	N
<i>Quercus alba</i>	26.6%	0.67	0.47	0.55	Tree	N
<i>Quercus coccinea</i>	15.7%	0.37	0.18	0.33	Tree	N
<i>Quercus falcata</i>	6.5%	0.52	0.18	0.35	Tree	N
<i>Quercus geminata</i>	2.3%	0.21	0.04	0.19	Tree, Shrub	N
<i>Quercus</i>						
<i>hemisphaerica</i>	7.0%	0.43	0.18	0.39	Tree	N
<i>Quercus incana</i>	6.6%	0.31	0.11	0.26	Tree, Shrub	N
<i>Quercus laevis</i>	8.0%	0.30	0.12	0.29	Tree	N
<i>Quercus laurifolia</i>	8.0%	0.45	0.15	0.32	Tree	N
<i>Quercus lyrata</i>	4.0%	0.24	0.05	0.24	Tree	N
<i>Quercus margaretta</i>	5.5%	0.32	0.11	0.23	Tree, Shrub	N
<i>Quercus marilandica</i>	8.4%	0.42	0.20	0.31	Tree, Shrub	N
<i>Quercus michauxii</i>	5.2%	0.35	0.12	0.24	Tree	N
<i>Quercus minima</i>	0.8%	0.18	0.02	0.06	Shrub	N
<i>Quercus montana</i>	29.0%	0.47	0.38	0.45	Tree	N
<i>Quercus nigra</i>	9.3%	0.62	0.27	0.47	Tree	N
<i>Quercus pagoda</i>	4.1%	0.40	0.11	0.22	Tree	N
<i>Quercus phellos</i>	6.1%	0.53	0.17	0.41	Tree	N
<i>Quercus rubra</i>	43.2%	0.62	0.57	0.54	Tree	N
<i>Quercus shumardii</i>	2.3%	0.29	0.04	0.16	Tree, Shrub	N
<i>Quercus stellata</i>	6.8%	0.56	0.21	0.37	Tree	N
<i>Quercus velutina</i>	19.1%	0.64	0.38	0.50	Tree	N
<i>Quercus virginiana</i>	3.1%	0.33	0.06	0.24	Tree	N
<i>Rhododendron</i>						
<i>atlanticum</i>	1.9%	0.24	0.04	0.15	Shrub	N
<i>Rhododendron</i>						
<i>calendulaceum</i>	13.2%	0.33	0.21	0.30	Shrub	N
<i>Rhododendron</i>	4.1%	0.23	0.06	0.24	Tree, Shrub	N

<i>catawbiense</i>						
<i>Rhododendron maximum</i>	27.3%	0.41	0.33	0.42	Tree, Shrub	N
<i>Rhododendron periclymenoides</i>	0.9%	0.16	0.01	0.12	Shrub	N
<i>Rhododendron viscosum</i>	1.8%	0.32	0.03	0.22	Shrub	N
<i>Rhus aromatica</i>	0.6%	0.18	0.01	0.12	Shrub	N
<i>Rhus copallinum</i>	11.5%	0.63	0.33	0.43	Tree, Shrub	N
<i>Ribes cynosbati</i>	2.4%	0.21	0.04	0.18	Shrub	N
<i>Ribes rotundifolium</i>	1.7%	0.16	0.03	0.18	Shrub	N
<i>Robinia hispida</i>	0.8%	0.09	0.01	0.12	Tree, Shrub	I
<i>Robinia pseudoacacia</i>	25.8%	0.53	0.41	0.45	Tree	I
<i>Rosa carolina</i>	2.2%	0.38	0.06	0.23	Subshrub	N
<i>Rosa multiflora</i>	1.6%	0.33	0.04	0.18	Vine, Subshrub	I
<i>Rosa palustris</i>	0.9%	0.18	0.00	0.15	Subshrub	N
<i>Rubus allegheniensis</i>	4.6%	0.31	0.11	0.26	Subshrub	N
<i>Rubus argutus</i>	7.7%	0.72	0.26	0.52	Subshrub	I
<i>Rubus canadensis</i>	14.7%	0.37	0.27	0.40	Subshrub	N
<i>Rubus cuneifolius</i>	2.0%	0.31	0.05	0.18	Subshrub	N
<i>Rubus hispidus</i>	1.0%	0.29	0.02	0.21	Subshrub	N
<i>Rubus odoratus</i>	0.7%	0.16	0.02	0.10	Subshrub	N
<i>Rubus trivialis</i>	1.5%	0.38	0.05	0.23	Subshrub, Vine	N
<i>Rudbeckia laciniata</i>	2.1%	0.35	0.07	0.19	Subshrub, Forb/herb	N
<i>Sabal minor</i>	2.4%	0.26	0.06	0.19	Tree, Shrub	N
<i>Sabal palmetto</i>	1.4%	0.13	0.02	0.12	Tree	N
<i>Salix nigra</i>	0.5%	0.19	0.00	0.11	Tree	N
<i>Sambucus racemosa</i>	3.0%	0.16	0.04	0.20	Tree, Shrub	N
<i>Sarracenia flava</i>	2.4%	0.24	0.04	0.17	Subshrub, Forb/herb	N
<i>Sarracenia minor</i>	1.7%	0.22	0.04	0.14	Subshrub, Forb/herb	N
<i>Sarracenia purpurea</i>	1.2%	0.17	0.03	0.11	Subshrub, Forb/herb	N
<i>Sassafras albidum</i>	33.9%	0.74	0.55	0.76	Tree, Shrub	N
<i>Scrophularia marilandica</i>	0.5%	0.24	0.00	0.12	Subshrub, Forb/herb	N
<i>Serenoa repens</i>	1.6%	0.20	0.03	0.14	Tree, Shrub	N
<i>Smilax auriculata</i>	1.7%	0.27	0.03	0.26	Shrub, Vine	N
<i>Smilax biltmoreana</i>	3.6%	0.26	0.07	0.25	Vine, Forb/herb	N
<i>Smilax bona-nox</i>	15.6%	0.71	0.38	0.69	Shrub, Vine	N
<i>Smilax glauca</i>	44.7%	0.91	0.83	0.94	Shrub, Vine	N
<i>Smilax hispida</i>	5.0%	0.41	0.10	0.37	Shrub, Vine	N
<i>Smilax laurifolia</i>	8.2%	0.47	0.18	0.48	Shrub, Vine	N
<i>Smilax pulverulenta</i>	0.9%	0.20	0.03	0.11	Vine, Forb/herb	N
<i>Smilax pumila</i>	0.9%	0.20	0.02	0.17	Shrub, Subshrub, Vine	N
<i>Smilax rotundifolia</i>	47.5%	0.91	0.77	1.00	Shrub, Vine	N
<i>Smilax smallii</i>	1.6%	0.22	0.03	0.17	Shrub, Vine	N
<i>Smilax walteri</i>	2.9%	0.29	0.05	0.31	Shrub, Vine	N
<i>Solanum carolinense</i>	0.6%	0.29	0.03	0.16	Subshrub, Forb/herb	I
<i>Sorbus americana</i>	3.2%	0.18	0.05	0.23	Tree, Shrub	N
<i>Staphylea trifolia</i>	0.5%	0.12	0.01	0.08	Tree, Shrub	N
<i>Stewartia ovata</i>	0.7%	0.10	0.01	0.09	Tree, Shrub	N

<i>Strophostyles</i>						
<i>umbellata</i>	0.7%	0.18	0.02	0.10	Vine, Forb/herb	N
<i>Stylisma humistrata</i>	0.7%	0.17	0.01	0.11	Vine, Forb/herb	N
<i>Stylisma patens</i>	2.3%	0.21	0.05	0.23	Vine, Forb/herb	N
<i>Styrax americanus</i>	0.5%	0.20	0.01	0.14	Tree, Shrub	N
<i>Styrax grandifolius</i>	0.9%	0.21	0.02	0.12	Tree, Shrub	N
<i>Symphoricarpos</i>						
<i>orbiculatus</i>	1.2%	0.26	0.04	0.16	Shrub	I
<i>Symplocos tinctoria</i>	11.6%	0.55	0.26	0.54	Tree, Shrub	N
<i>Taxodium ascendens</i>	2.2%	0.22	0.04	0.19	Tree	N
<i>Taxodium distichum</i>	4.7%	0.23	0.06	0.27	Tree	N
<i>Tephrosia virginiana</i>	6.8%	0.41	0.16	0.29	Subshrub, Forb/herb	N
<i>Tilia americana</i>	15.7%	0.45	0.30	0.32	Tree	I
<i>Tillandsia usneoides</i>	6.7%	0.48	0.14	0.49	Forb/herb, Vine	I
<i>Toxicodendron</i>						
<i>pubescens</i>	5.6%	0.36	0.11	0.27	Forb/herb	N
<i>Toxicodendron</i>						
<i>radicans</i>	27.9%	0.93	0.69	0.82	Shrub, Forb/herb, Subshrub, Vine	N
<i>Trachelospermum</i>						
<i>difforme</i>	2.6%	0.36	0.07	0.29	Vine	N
<i>Tragia urticifolia</i>	0.9%	0.24	0.01	0.12	Vine, Forb/herb	N
<i>Triadica sebifera</i>	0.5%	0.16	0.01	0.10	Tree	I
<i>Tsuga canadensis</i>	33.5%	0.43	0.39	0.41	Tree	N
<i>Tsuga caroliniana</i>	2.9%	0.21	0.05	0.21	Tree	N
<i>Ulmus americana</i>	8.1%	0.44	0.02	0.09	Tree	N
<i>Ulmus rubra</i>	5.0%	0.49	0.15	0.38	Tree	N
<i>Vaccinium arboreum</i>	7.1%	0.53	0.19	0.45	Tree, Shrub	N
<i>Vaccinium</i>						
<i>corymbosum</i>	10.3%	0.34	0.16	0.37	Shrub	N
<i>Vaccinium</i>						
<i>crassifolium</i>	5.9%	0.27	0.11	0.20	Shrub, Subshrub	N
<i>Vaccinium elliotii</i>	2.1%	0.34	0.05	0.23	Shrub	N
<i>Vaccinium</i>						
<i>erythrocarpum</i>	5.5%	0.21	0.10	0.29	Shrub	N
<i>Vaccinium formosum</i>	7.4%	0.45	0.18	0.41	Shrub	N
<i>Vaccinium fuscum</i>	6.3%	0.44	0.14	0.41	Shrub	N
<i>Vaccinium hirsutum</i>	2.7%	0.11	0.03	0.10	Shrub	N
<i>Vaccinium myrsinites</i>	2.0%	0.21	0.04	0.13	Shrub	N
<i>Vaccinium pallidum</i>	19.5%	0.55	0.31	0.54	Subshrub, Shrub	N
<i>Vaccinium simulatum</i>	4.8%	0.27	0.09	0.25	Shrub	N
<i>Vaccinium stamineum</i>	20.2%	0.66	0.42	0.67	Shrub	N
<i>Vaccinium tenellum</i>	12.8%	0.50	0.30	0.32	Subshrub, Shrub	N
<i>Vaccinium virgatum</i>	0.7%	0.22	0.03	0.14	Shrub	N
<i>Viburnum dentatum</i>	0.2%	0.10	0.07	0.28	Tree, Shrub	N
<i>Viburnum lantanoides</i>	5.2%	0.18	0.04	0.23	Shrub	N
<i>Viburnum nudum</i>	1.6%	0.28	0.07	0.20	Tree, Shrub	N
<i>Viburnum obovatum</i>	0.2%	0.11	0.02	0.21	Tree, Shrub	N
<i>Viburnum rufidulum</i>	1.6%	0.29	0.03	0.11	Tree, Shrub	N
<i>Vicia caroliniana</i>	0.8%	0.23	0.03	0.13	Vine, Forb/herb	N
<i>Vitis aestivalis</i>	13.1%	0.65	0.32	0.53	Vine	N

<i>Vitis cinerea</i>	3.2%	0.43	0.09	0.28	Vine	N
<i>Vitis labrusca</i>	1.3%	0.31	0.03	0.20	Vine	I
<i>Vitis rotundifolia</i>	26.0%	0.83	0.55	0.77	Vine	N
<i>Vitis vulpina</i>	1.7%	0.27	0.04	0.21	Vine	N
<i>Xanthorhiza simplicissima</i>	0.7%	0.22	0.02	0.11	Subshrub, Forb/herb Subshrub, Shrub,	N
<i>Yucca filamentosa</i>	0.7%	0.22	0.02	0.16	Forb/herb	N
<i>Zenobia pulverulenta</i>	1.0%	0.11	0.02	0.07	Shrub	N

DISCUSSION

Performance across scenarios

The performance of all co-occurrence based metrics depends greatly on the nature of data being evaluated, including the size and distribution of species niches in niche space, and the sampling method used. Substantially biased site selection decreased performance by about 0.05, and this was not ameliorated by random sub-sampling of the data. In fact, by reducing the number of co-occurers considered at a time, sub-sampling decreased performance, especially if normalization of abundances is used. While sub-sampling may, therefore, be useful in some extreme situations and with some metrics, it cannot be recommended generally.

It is not surprising that the best estimates were achieved when species niche optima were equally distributed along an environmental gradient. This is because all metrics use the appearance of non-focal species as milestones representing niche space. If niches are clustered, the niche-space distance represented by those milestones varies along the gradient, making estimates worse by about 0.10; while a uniformly random distribution is somewhat less problematic, as long as the number of species is not very small. These losses in performance are not so large that they remove all utility from the metrics—even

when a researcher has little knowledge of how the niches of his study taxa are distributed.

The distribution of niche widths in a pool of species is even more important. It makes sense that scenarios with narrow, but not extremely narrow, niches created better estimates: if niches are too small, richness is low and any focal species will have few co-occurers; conversely, if they approach the length of the sampled gradient, much of each niche will be unmeasurable and all species will appear similar. Co-occurrence-based metrics should not be applied either to monocultures or areas where turnover is nonexistent. As long as niches are wide enough to commonly overlap, it may be that narrow ones are preferable because the average distance from any point in the niche is closer to the optima, meaning that co-occurrence with that species is more likely to mean occurrence at its optima; and it is optima that are more reliably spaced along the gradient, and so serve as better milestones. (If niches were perfectly spaced and their widths were identical, there would be no meaningful difference.)

Similar reasoning explains the apparent benefit of niche widths being drawn from a gamma distribution (producing a “hockey-stick” curve) instead of a uniform one. Given the same mean, most niches are narrow, allowing reliable measurement of optima; yet the narrow peak of the distribution produces a few generalists, so that few species are without co-occurers. One consequence of this may lie in the selection of taxa a researcher considers. For instance, if trying to assess the niche widths of a single genus of plants, he might include species outside that genera, to ensure some generalists are counted and site richness is never extremely low.

Metric Choices

Additional Uses of the Data

The first question (A) regarding alternative metric construction methods was the utility of including data beyond species presence.

The combinatoric methods, considering the number of unique assemblages each focal species and co-occurrence inhabit together, failed this test; in no set of scenarios did performance improve. This may be because absences are thus counted equally with presence: an assemblage, minus one species, is counted as a separate assemblage—even when such an absence is not due to an inability of the missing species to exist, but is stochastic. Thus, the combinatoric methods are susceptible to both type I and type II errors, with the latter becoming worse with higher saturation. There is evidence for this explanation in the fact that subsetting (with normalization) is highly useful with combinatoric methods: logically, since it removes the harder-to-sample tails of species niches.

The inclusion of abundance, in contrast, was widely useful, yielding better performance than a simple count except when niche widths were uniformly distributed and wide. This makes sense as in such a case saturation will be quite high and local abundance (even more than presence) will be highly stochastic. When this is not the case, though, considering the maximum abundance seen for each co-occurrence makes the counting process more focused on optima. This focus is enhanced with a normalization of abundances, as typical niche height inherent in any species becomes irrelevant, so that

the maximum abundance only reveals amount over overlap. When normalization is used, maximum-abundance counting is generally the best method.

Subsetting and Normalization

In highly saturated sites both presence and abundance become less a result of abiotic suitedness and more stochastic forces. Ignoring low-abundance occurrences could restrict an analysis to more reliable data. Such subsetting harmed performance unless normalization was performed beforehand, so that inherently non-abundance rare species were not generally excluded. With normalization, light subsetting could mildly improve estimates. Obviously, this can only be undertaken when abundances are available in a dataset. It will also likely only be useful when a large portion of most species' ranges is included, so that normalization is accurately scaling abundances relative to what is possible. If the sample is small, subsetting could have a highly negative impact. The exact level of ideal subsetting is difficult to guess. Here, 0.2 was optimal, but this will depend on the level of saturation. Perhaps measures of turn-over could be informative.

Richness Correction

Some portions of a gradient may be particularly speciose, effectively changing units of niche measurement along its length. This can be countered by factoring out this richness, but the method depends on how saturated the sites tend to be; that is, whether local richness is representative of the local species pool size, or is independent. When sites were highly saturated, all richness correction schemes were counter-productive. This may be because when the pool is so large, variance in niche distributions does not

actually manifest as differences in richness. When saturations were lower, division was the best performer, as has been suggested before. Division performed by co-occurrence richness did slightly better than division by mean focal species richness when saturation was high, as did subtraction; but neither was an improvement on no action at all. Co-division was hypothesized to be more effective when richness changed drastically within a focal species' range, but it seems that other factors made this small difference moot.

Metric Selection

While some of the metrics tested are almost universally undesirable, it is clear that many would be useful in the right situation; that is, applied to the right taxon at the right point in time and space. However, most of the relevant community properties are difficult for researchers to assess accurately. The most readily identifiable "parameter" is likely the distribution of widths ("hockey-stick" or uniform) but the best-performing metric is the same in either case. Some estimate of saturation might also allow the subsetting threshold to be set (higher in the case of greater saturation).

The relatively small differences in metric performance likely do not warrant a very exhaustive effort to choose the right one, making a single metric that performs adequately across many scenarios preferable. #55n was identified as having the highest mean r^2 value and the highest minimum in any simulation run. It is not perfect having beat the JF07 metric (#2y) only 72% of the time, yet on average it did so by 0.08 r^2 . (It beat #1n, the simple co-occurrence count, 90% of the time by an average of 0.11). In terms

of its ready use, #55n is slightly harder to implement than #2y, but runs significantly faster since it does not include data sub-sampling. #55 thus seems the safest choice.

Conclusions

We should be cautious in using co-occurrence data to estimate species generalism as many properties of the community and the sampling regime affect performance. Even the best-performing metric showed a very large range of r^2 values in the 7,200 simulations it was tested in: at worst 0.11 and at best 0.98. Even when parameters were ideal performance went as low as 0.72, and in a real dataset we may have scant idea whether conditions are good for making estimates or not. If possible, though, we should ensure that our sample is large, to encompass as much environmental variation as possible, and unbiased, because sub-sampling does not uniformly offer an adequate corrective.

An inescapable limitation of these metrics is that they can only measure realized niche: chance, history and biotic interactions are all implicitly included in a species' score, which will thus vary in time and space. This is not entirely a disadvantage though, for it allows the measurement of niche changes in response to various ecological processes such as disturbance regime, invasions or climate change.

Of the 192 metric variants tested, #55n emerged as a good, safe choice; each species' local abundances should be normalized with a cumulative density function and values below 0.2 should be omitted. For every focal species, the maximum value of every co-occurrence should be found, and the total divided by mean richness where the focal

species appears. If the estimates thereby produced are analyzed conservatively, they should be of value to ecologists and biogeographers.

WORKS CITED

- Hutchinson, G. E. & MacArthur, R. H. (1959) A Theoretical Ecological Model of Size Distributions Among Species of Animals. *The American Naturalist*, **93**, 117–125.
- Fridley, J. D., Vandermaast, D. B., Kuppinger, D. M., Manthey, M., & Peet, R. K. (2007) Co-occurrence based assessment of habitat generalists and specialists: a new approach for the measurement of niche width. *Journal of Ecology*, **95**, 707–722.
- Højsgaard, S. (2009) doBy: Groupwise computations of summary statistics and other utilities. R package version 3.7.
- Manthey, M. & Fridley, J. D. (2008) Beta diversity metrics and the estimation of niche width via species co-occurrence data: reply to Zeleny. *Journal of Ecology*, **97**, 18–22.
- R Development Core Team (2009) R: A Language and Environment for Statistical Computing. Version 2.8.1.
- Peet, R. K., Wentworth, T. R., Schafale, M. P., & Weakley, A. S. (2004) Carolina Vegetation Survey database. Version 3.0. North Carolina Botanical Garden. Chapel Hill, NC 27599.
- Sarkar, D. (2009) lattice: Lattice Graphics. R package version 0.17-20.
- Vetaas, O. R. (2002) Realized and potential climate niches: a comparison of four *Rhododendron* tree species. *Journal of Biogeography*, **29**, 545–554.
- Zeleny, D. (2008) Co-occurrence based assessment of species habitat specialization is affected by the size of species pool: reply to. *Journal of Ecology*, **97**, 10–17.

2. The Multi-Grain Niche

Species generalism and sampling area

Introduction

Ecological patterns emerge at a variety of spatial scales, making scale of crucial interest to ecologists (Weins, 1989; Levins, 1992). For a given question, data are useful only if collected at a relevant scale (Reed et al. 1993), which must first be identified (e.g., Hurlbert, 2007). Species' distributions are limited partly by various environmental requirements (i.e., their niches in n-dimensional space). Different variables may tend to vary over different spatial scales, as do the processes behind them, leading to distributions that do the same. By identifying the scale at which a species is most limited (i.e., the scale at which its apparent niche is narrowest) we should be able to identify the scale of process most relevant to that species, and perhaps then the specific processes.

Such an investigation might seem wholly impractical if not tautological; to measure niche breadth at a given scale we must sample environmental variables at that scale. However, we do not know which variables are relevant enough to measure because this is exactly the question we are asking! Only by measuring all variables at all scales could we escape this bind. However, Fridley et al. (2007) showed that niche width could be estimated without measuring the environment—because species with many co-occurers tend to be more general, species occurrence data are sufficient.

My first goal was to identify the scale of processes that plants are most sensitive to, and the amount of variation present between species. Differences may be a product of life history strategies. For example, ruderal species which can be outcompeted within patches but which easily spread between them, should logically be less constrained at large scales. Common non-native species should also show high generalism at large scales, since without it they would not have become broadly established.

To assess generalism across scales, I applied a modified metric (McChesney, in draft) to data from the Carolina Vegetation Survey (Peet et al. 2004). This has been done previously but only with data at the full-plot scale of 1000 m². The CVS protocol was designed with multi-scale analysis in mind though, with each plot containing nested sub-modules at five levels from 0.01 m² to 1000 m² (Peet and Wentworth, 1998). County-level occurrence data from the USDA provided relative niche estimates at a larger scale.

Reliability

It is not immediately clear that niche estimates from different scales can reasonably be compared owing to two potential problems. First, the metric's efficacy has been tested through simulation modeling, but the parameters used may not obtain at some or all real-world grain sizes. Small grains in CVS cover less space and therefore provide a smaller sample of data which is less likely to be representative. Large grains effectively loosen the definition of co-occurrence, which will inflate all estimates, especially in heterogeneous landscapes. Second, even if estimates are equally good at all scales, they are technically calculated in different units, so may not be comparable without some mathematical correction. Therefore, preceding the analysis of real species, I sought to

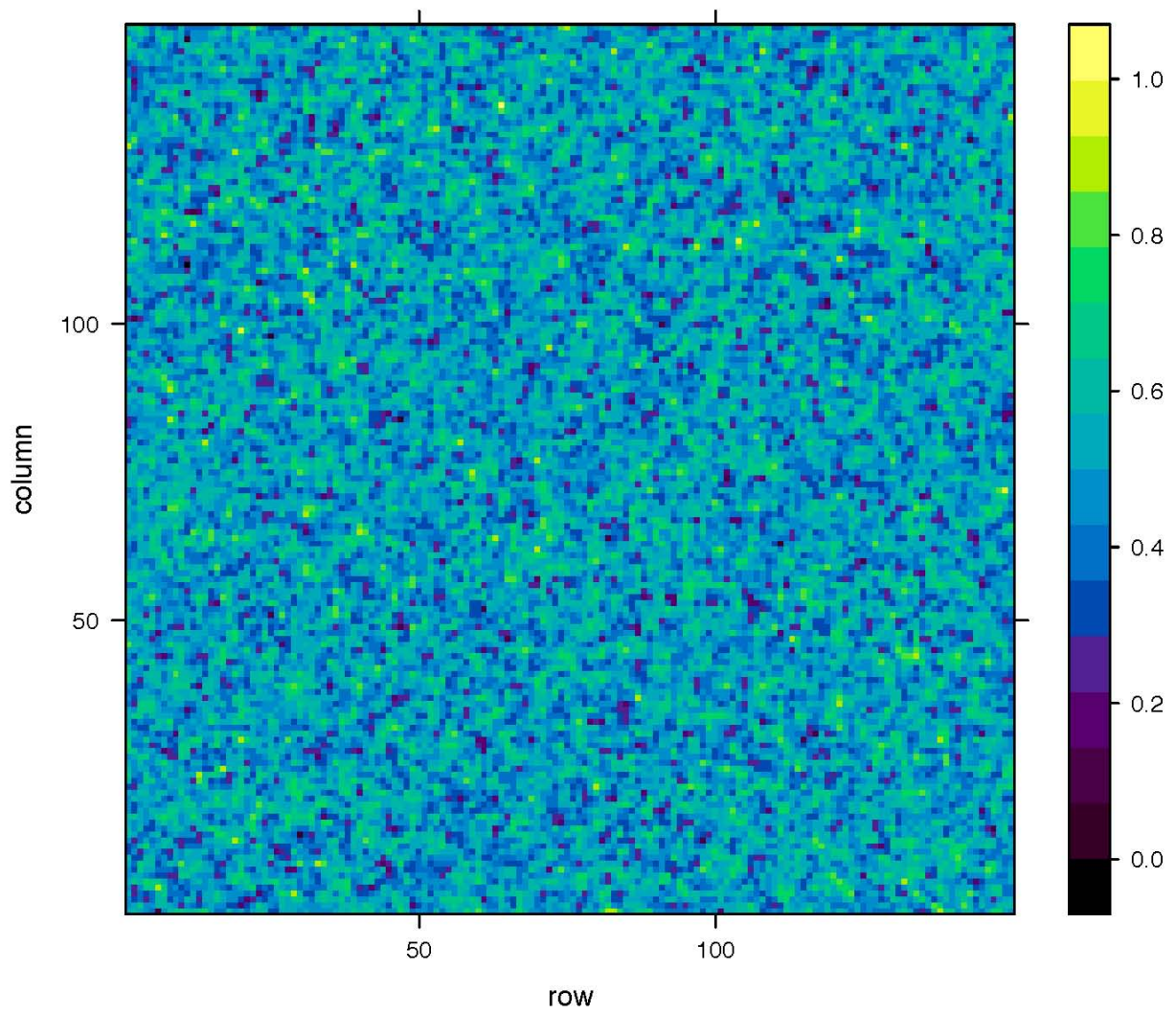
characterize metric operation across scales via simulation modeling. I looked for differences in metric performance and for biases in the estimates—as well as ways to compensate for these.

Methods

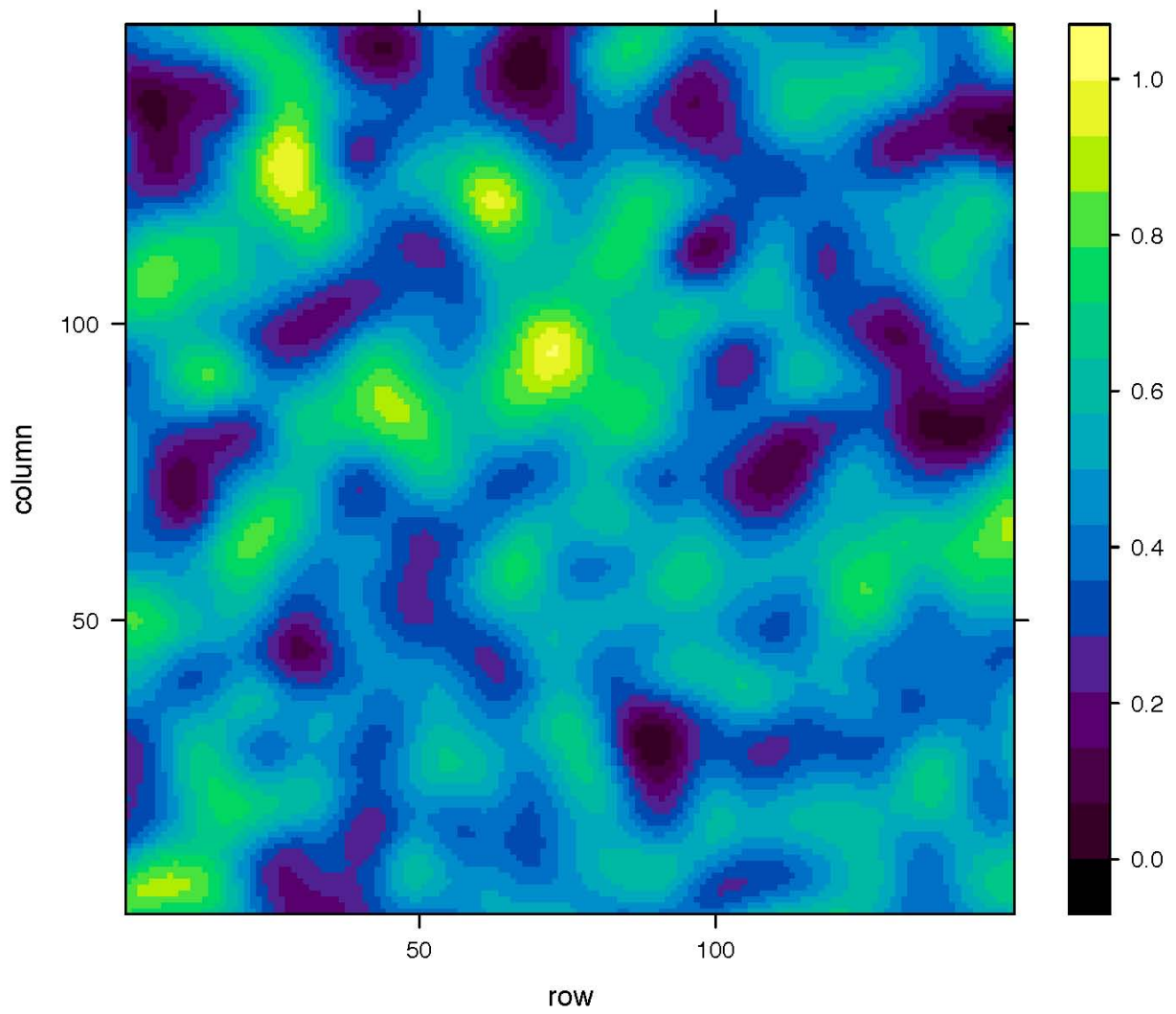
Simulation Model

To investigate the utility of Fridley et al.'s (2007) theta metric across scales, I applied it to fully known species niches and environments. These were created with a simulation model adapted from that used in McChesney (in draft). Normal-shaped species response curves were randomly created on a single resource axis. A two-dimensional grid modeled the values of that resource in space: the variation was generated with Gaussian-smoothed noise (Schlather, 2009) where heterogeneity could be controlled: Fig. 2-1A–C shows a 150x150 grid such as was used, with smoothing at scales of 1, 10 and 100.

Smoothing of 1



Smoothing of 10



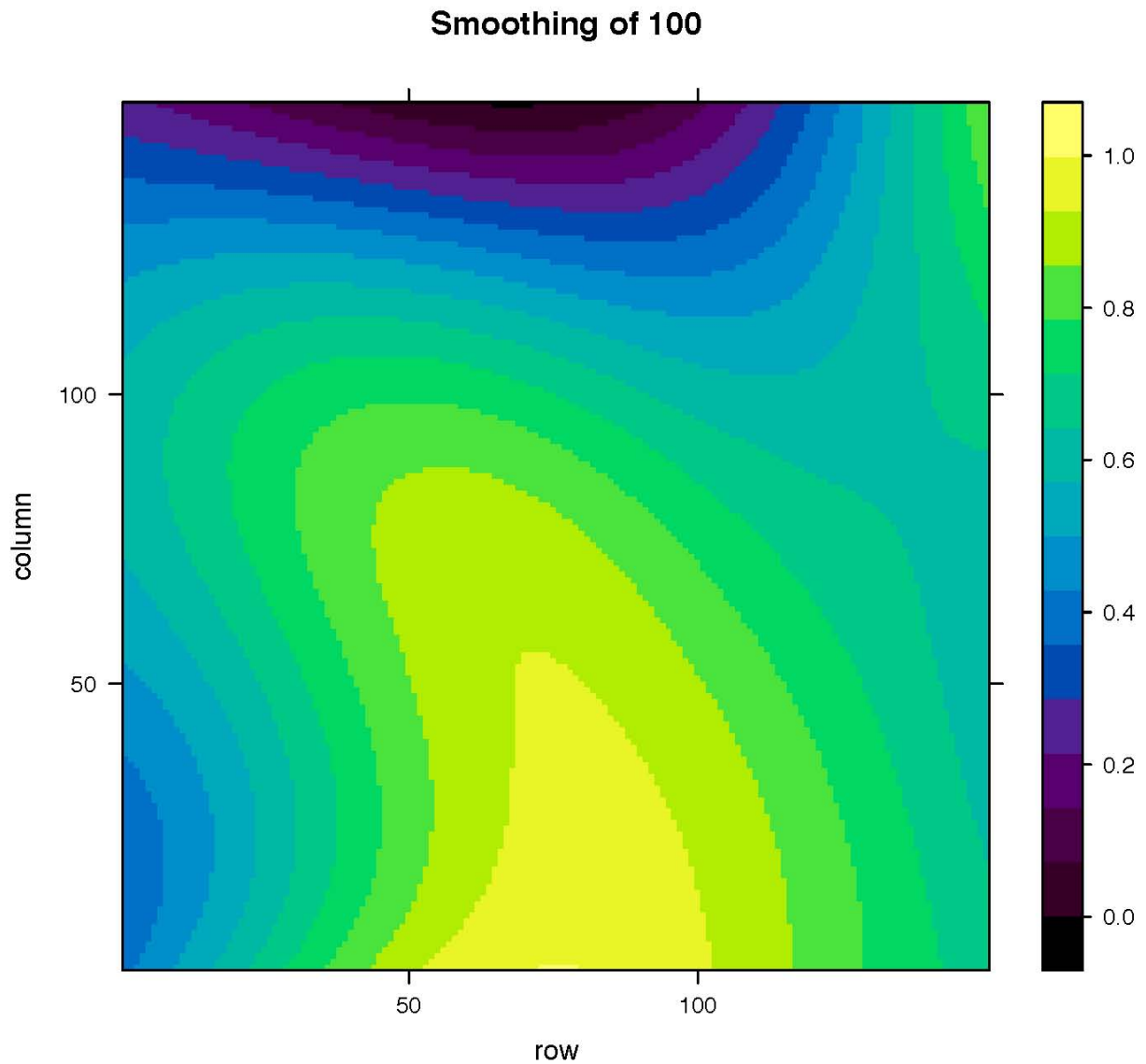


Fig. 2-1. Randomly generated landscapes with different amounts of Gaussian smoothing.

Each cell was assigned one individual drawn from the species pool: the probability of a species being selected was equal to the height of its response curve at the given resource value.

Variability in grain size was implemented by clustering grid cells into blocks, which tiled the landscape without overlap. The presence or absence of each species was tallied per block; no abundance data were generated. A number of blocks were then chosen randomly to constitute the sample, with data from these fed to the metric.

Niches were estimated at multiple grains in each simulation run. Multiple simulations were run with varying resource heterogeneity, with 100 replicates for each. Resource heterogeneity and grain values were increased exponentially rather than linearly because trial runs suggested that output is more sensitive to variation at the low end of these parameters.

Table 2-1. Simulation parameters.

parameter	Values
n species	50
mean niche width	0.2
n total plots	100
resource scales	1, 3, 10, 30, 100
grains	4, 9, 25, 49, 100, 196
spp. per cell	1

Carolina Vegetation Survey data

I obtained the all plots in the CVS database as of May, 2009 that were 1000 m² and included at least four 100 m² modules and all “nest” levels, down to 0.01 m² (n = 357). Within this subset were 1,157 higher plants from 138 families. Sub-species and variety records were aggregated to the species level. Records unknown to the species were

omitted. The data from each sampling level were converted to a presence-absence matrix; with cover estimates omitted, as they are not available below the module level.

Because each plot contains multiple modules, matrices for sub-plot grains contained more location entries than those at the plot grain. This introduces the issue of pseudo-replication, as modules from the same plot are spatially auto-correlated. However, although they contain more entries, these matrices actually contain less data in one sense, as they sample less physical space. I counted this latter problem the greater one, and therefore included all data from each sampling grain, treating each module as its own location.

To extend the range of sampling grains considered, county-level occurrences were obtained from the USDA PLANTS database (2009). Although county boundaries are not uniform or square, the data were considered to be taken at a $1.394 \times 10^9 \text{ m}^2$ grain, the average county size.

Width estimation

A modified version of the theta metric from Fridley et al. (2007) was used to estimate species niche widths at each grain. The variant, "#55n," was put forth in McChesney (in draft) and selected because of good performance across a range of scenarios. The relationship between sampling area and theta was then summarized with a linear model for each species, and comparisons were made across categories of interest, i.e., family, native status and growth form, as in PLANTS (USDA, 2009).

RESULTS

Grain and metric performance

Metric performance varied with grain and resource heterogeneity. In general, the best estimates were obtained in homogenous landscapes (Fig. 2-2). In the most heterogeneous ones (smoothing = 1) performance was best with small sampling grain, whereas middling and large grains did better in homogenous ones.

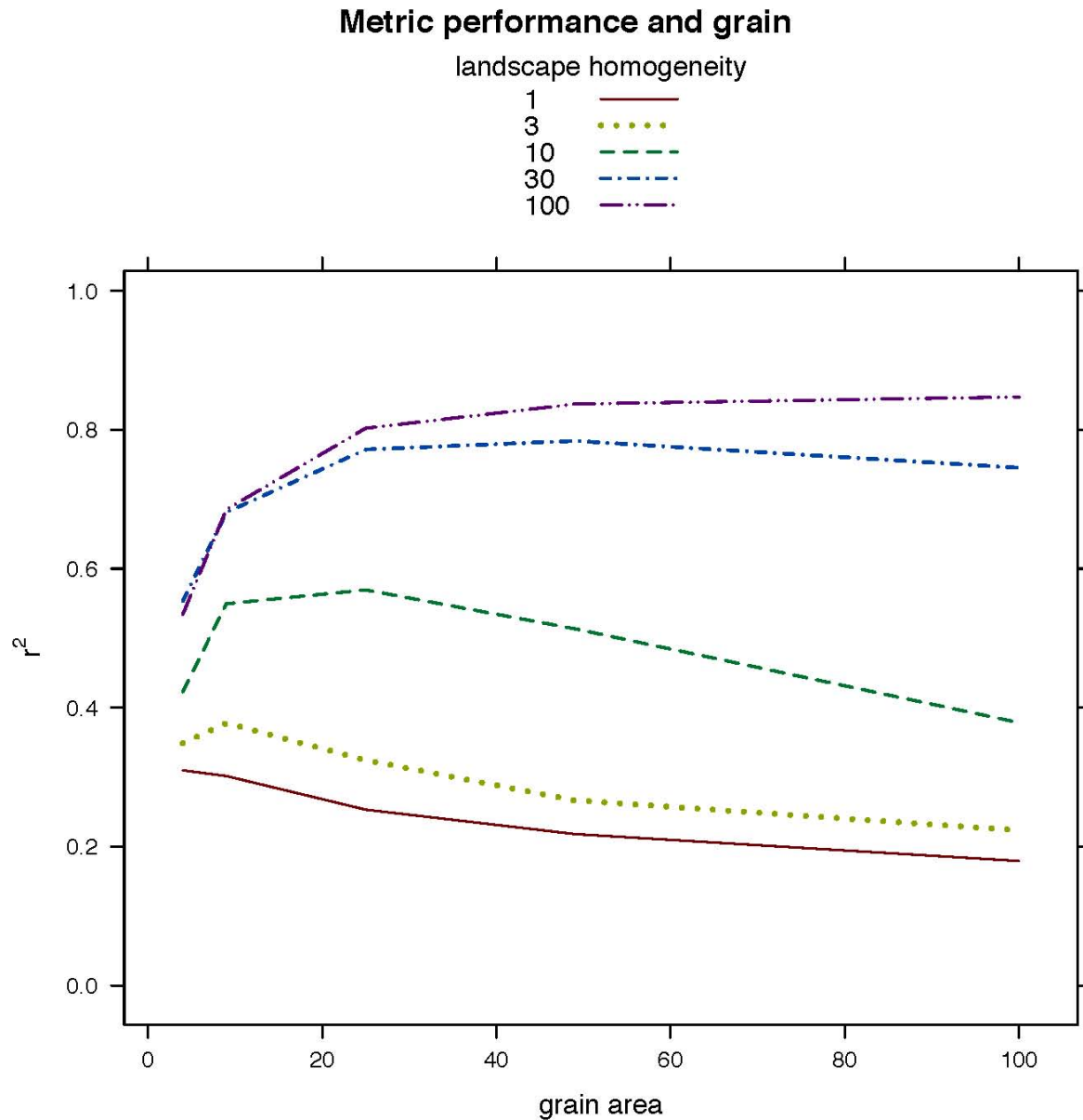


Fig. 2-2. Performance with different sampling grains. Lines show scale of spatial heterogeneity (inverse auto-correlation). Panels show niche overlap.

Landscape heterogeneity and small sample grain both contributed to in-plot resource variation, which was defined as the range of resource values present (Fig. 2-3). Looking across all simulations, this in-plot variation was strongly correlated with metric performance (Fig. 2-4; ANOVA showed r^2 of 0.459, $p < 2.2 \times 10^{-16}$).

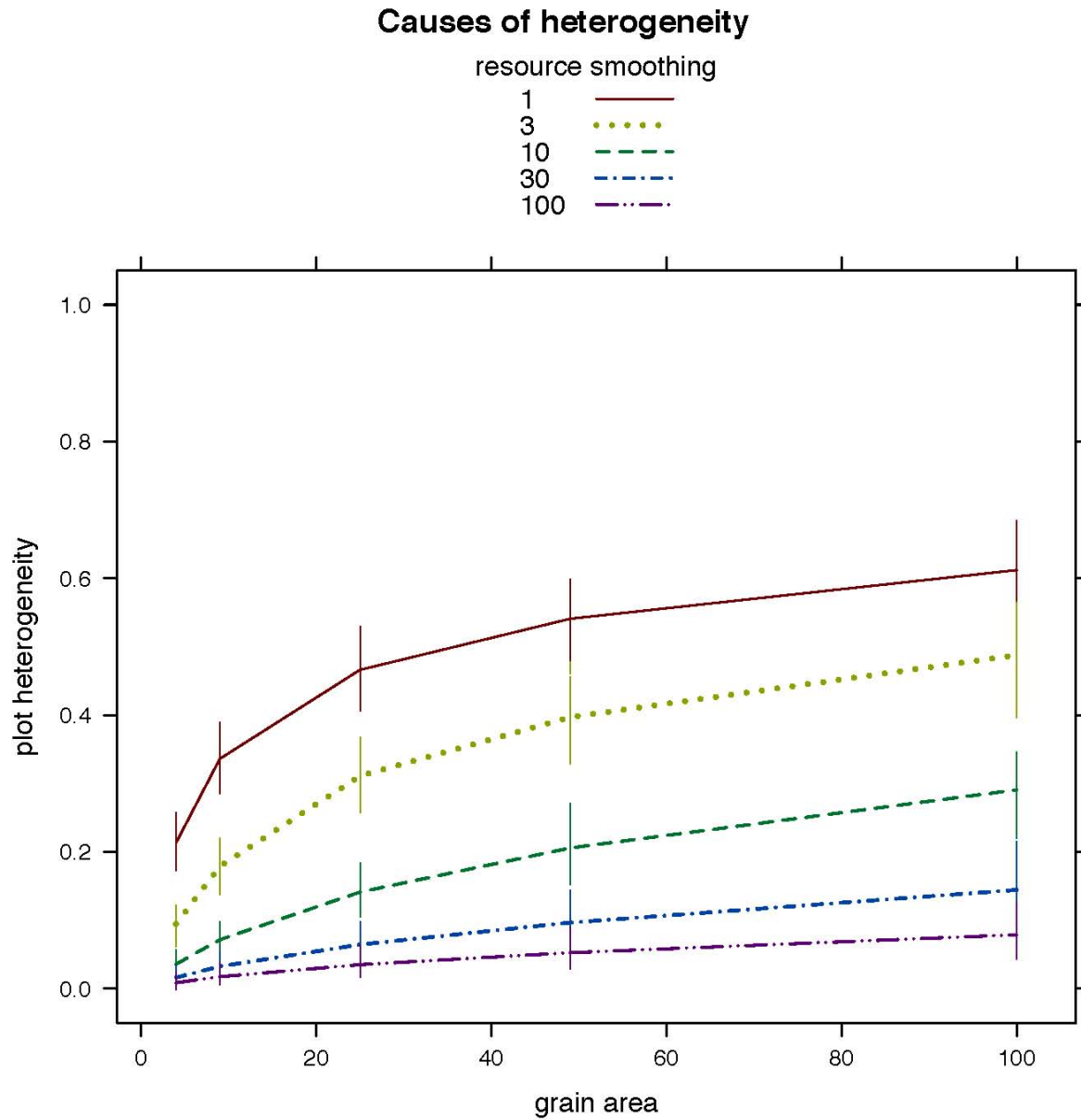


Fig. 2-3. Causes of plot heterogeneity, measured as the range in resource levels contained in the average plot at the given grain. Each point shows one replicate while lines show averages at different scales of resource variation (i.e. degree of smoothing).

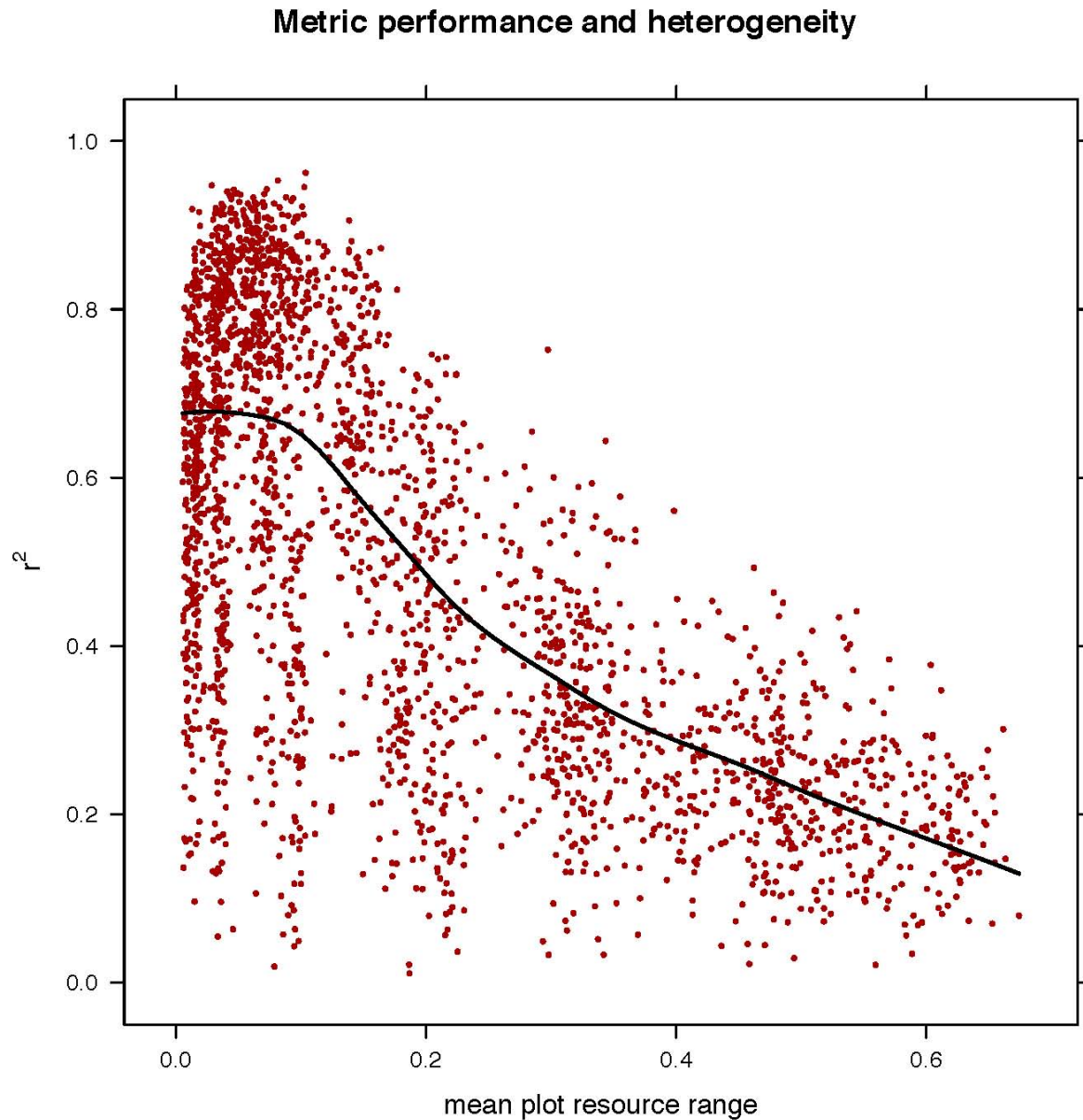


Fig. 2-4. Effect of plot heterogeneity on metric performance.

The grain producing the best niche estimates was identified for each type of landscape. This varied substantially between replicates; that is, the best grain depended on the particular configuration of species utilization curves in niche space, as well as the selection of plots from the entire landscape (which here were random). Nonetheless,

homogenous landscapes, which vary at a large scale, were best assessed with large-grained sampling, and vice-versa (Fig. 2-5; ANOVA showed r^2 of 0.508, $p < 2.2 \times 10^{-16}$).

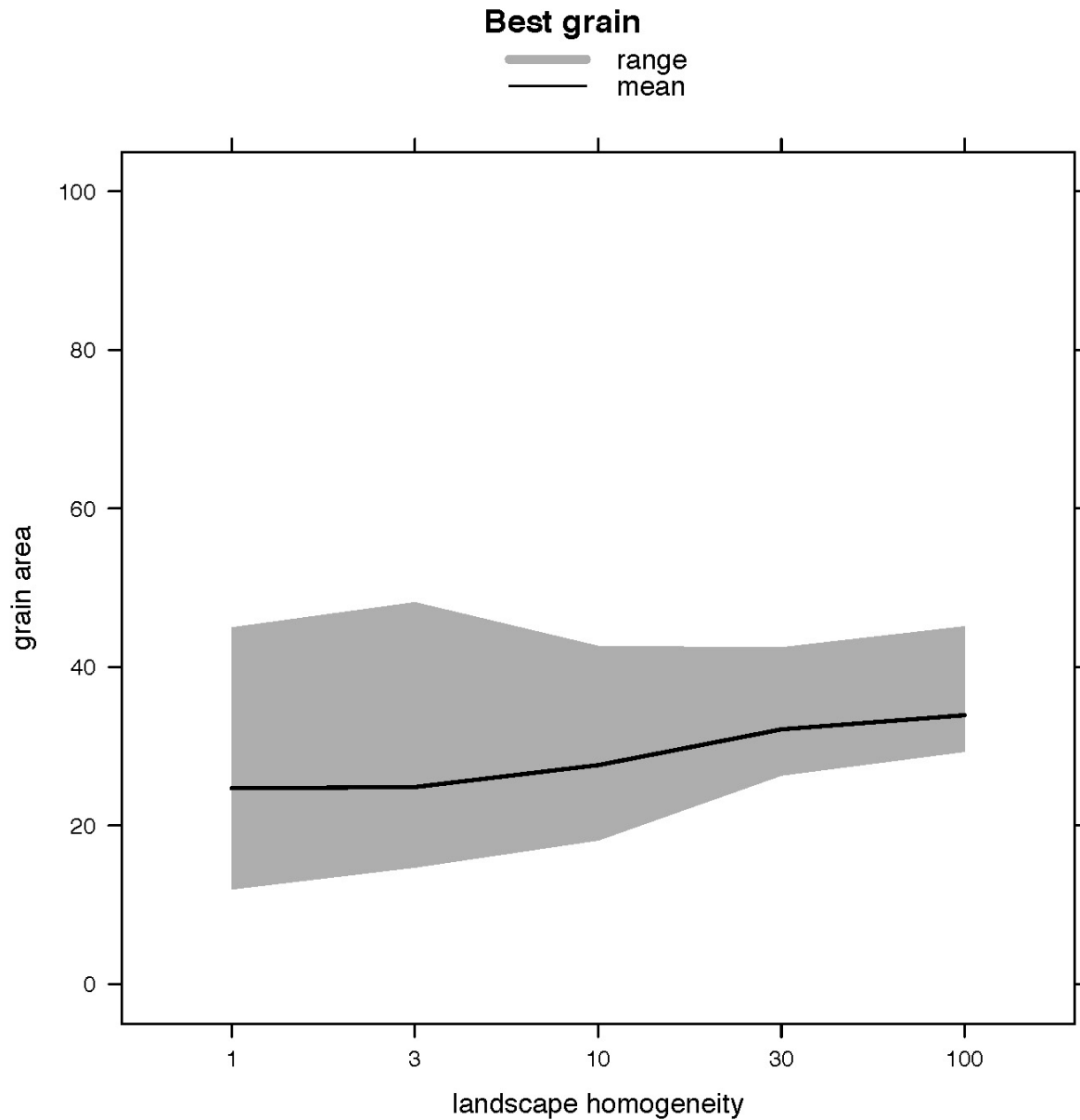


Fig. 2-5. Most suited grain (ordinate) for measuring niche widths at the given level of spatial heterogeneity (abscissa). Shaded area shows the range of selected grains in 100 replicates; solid line shows the mean.

However, in the real world it may be difficult to assess the scale of environmental turnover, especially because multiple limiting resources may be involved that vary at different scales. The best performing grains were often those that generated the greatest range in niche widths—which is useful knowledge because while performance can never be known in the real world, variation in estimates can be (Fig. 2-6).

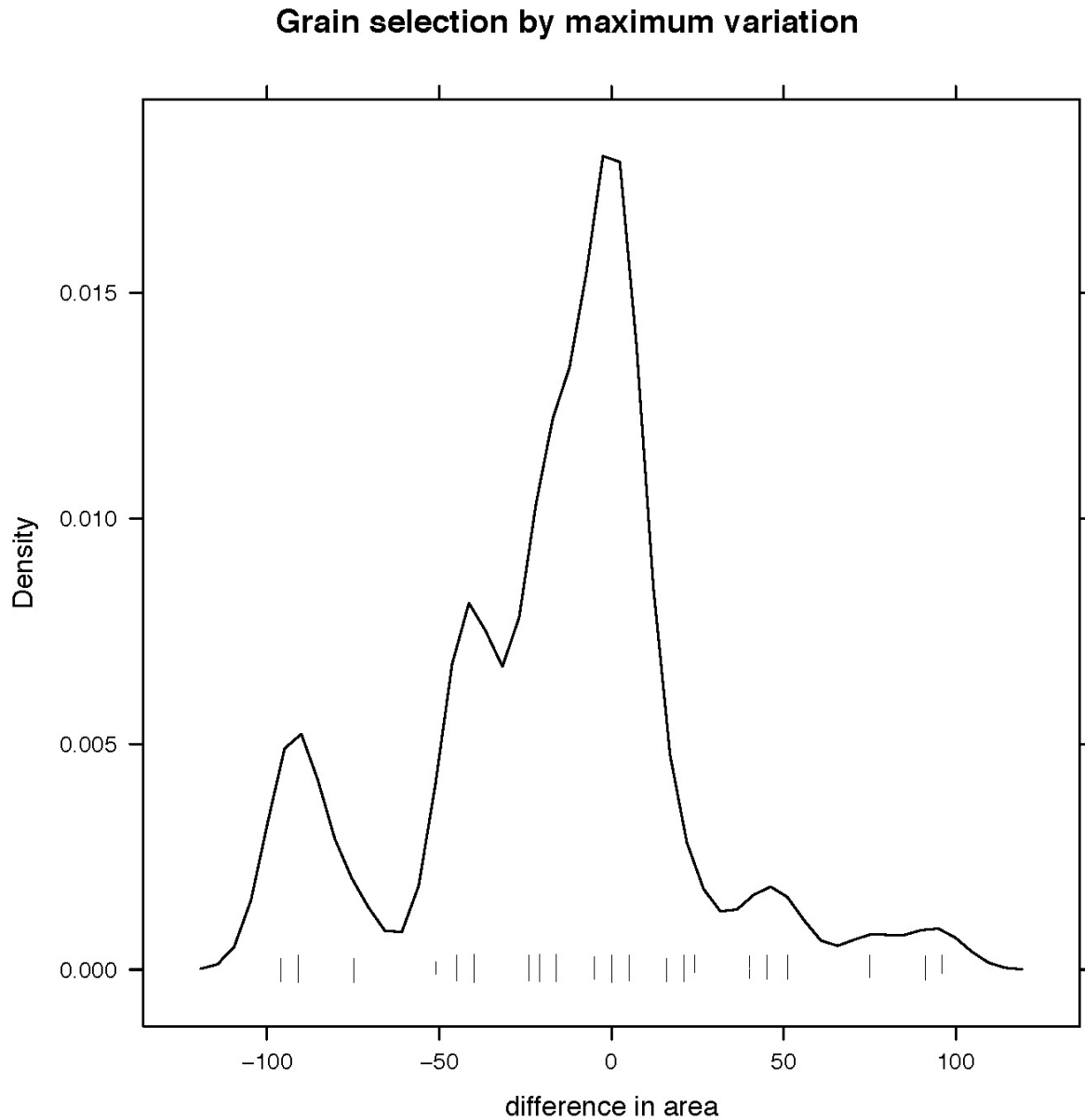


Fig. 2-6. Density plot showing the difference, in area, between the best performing grain in each scenario and the grain having the greatest variation in its species' niche width estimates. Because the difference is often small, those two grains are, in fact, the same. Vertical ticks represent data points (i.e. simulations).

Grain and estimate bias

Ideally, niche estimates should vary little with grain, because only one resource is modeled and in each simulation run it varies across the landscape at one scale; i.e. the

realistic complexity of a fractally patchy landscape is absent. However, niche estimates did vary with sample grain, and showed peak values near the relatively small grain of 9 (Fig. 2-7). This pattern was maintained in all landscapes, but sharpest in heterogenous ones (with smoothing of 1, 3 or 10).

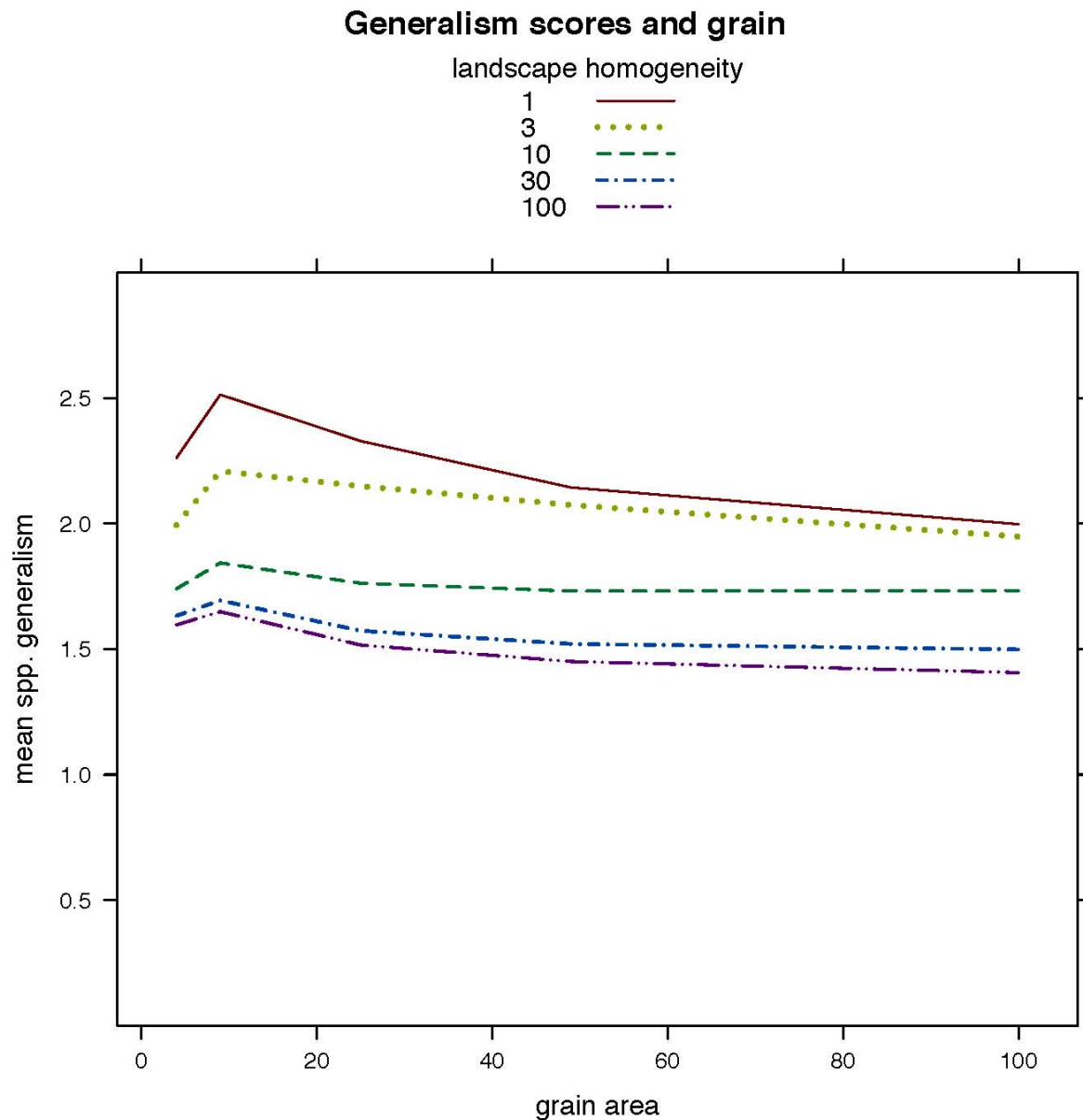


Fig. 2-7. Change in mean generalism score as sampling grain varies. Lines show different smoothing levels applied to resource distribution in space (i.e., scale of heterogeneity).

To correct for this lack of constancy between grains, several normalization techniques were tested. Their effectiveness was quantified with per-species variation in niche estimate, within in each replicate, which ideally would be 0 but in the unmodified data averaged 0.30 and was particularly high in heterogeneous scenarios (Fig. 2-8). Each technique was applied to all estimates for a particular replicate and grain. Dividing estimates by the mean estimate value reduced average variation from 0.30 to 0.08. Linear scaling of values (in which the maximum becomes 1 and the minimum 0) reduced variation to 0.03, though it was much higher in the heterogenous landscapes. Lastly, an empirical cumulative density function transformed each value into a quantile; this reduced variation to 0.02. The ECDF was therefore used in the analysis of real-world data.

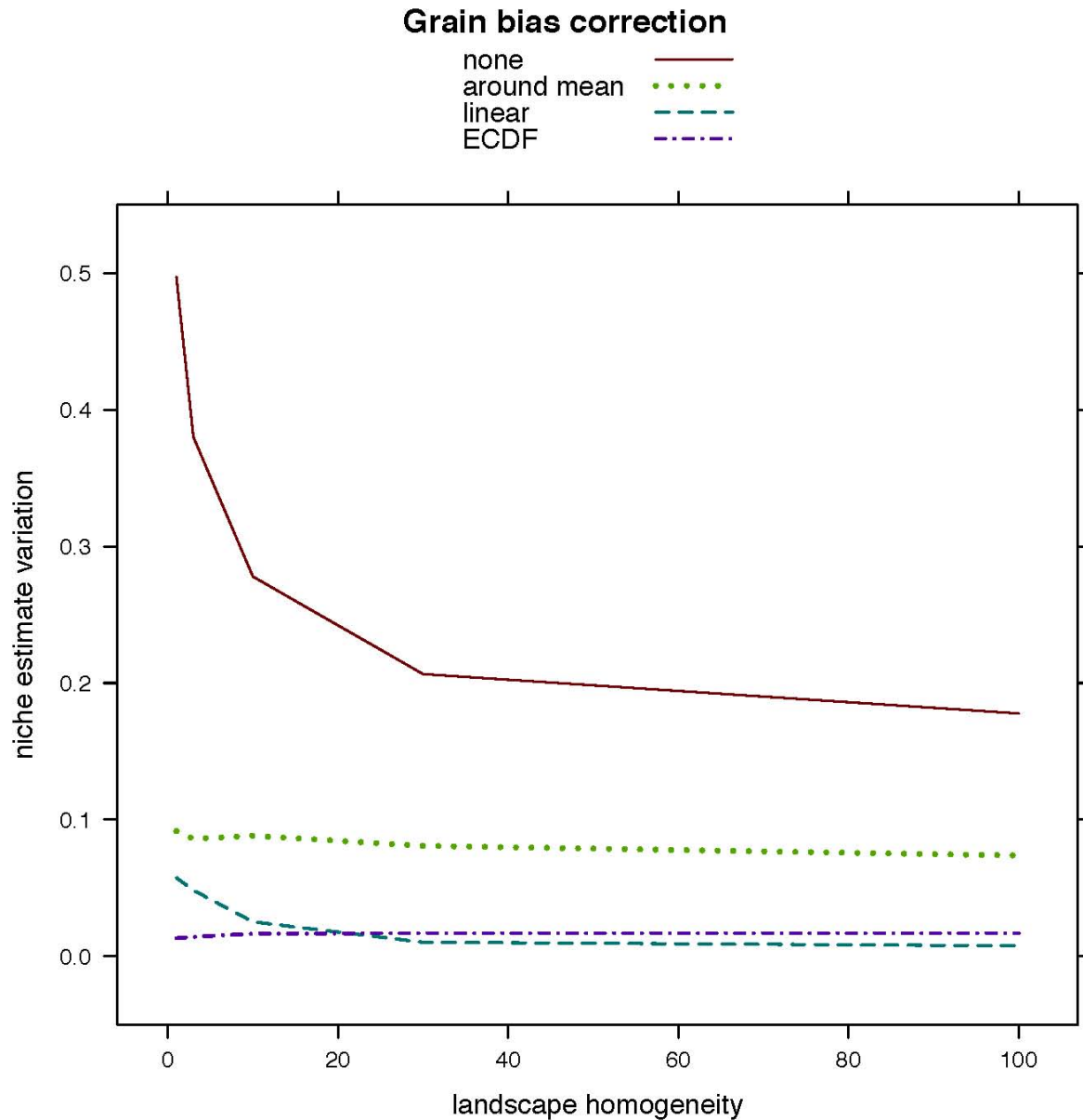
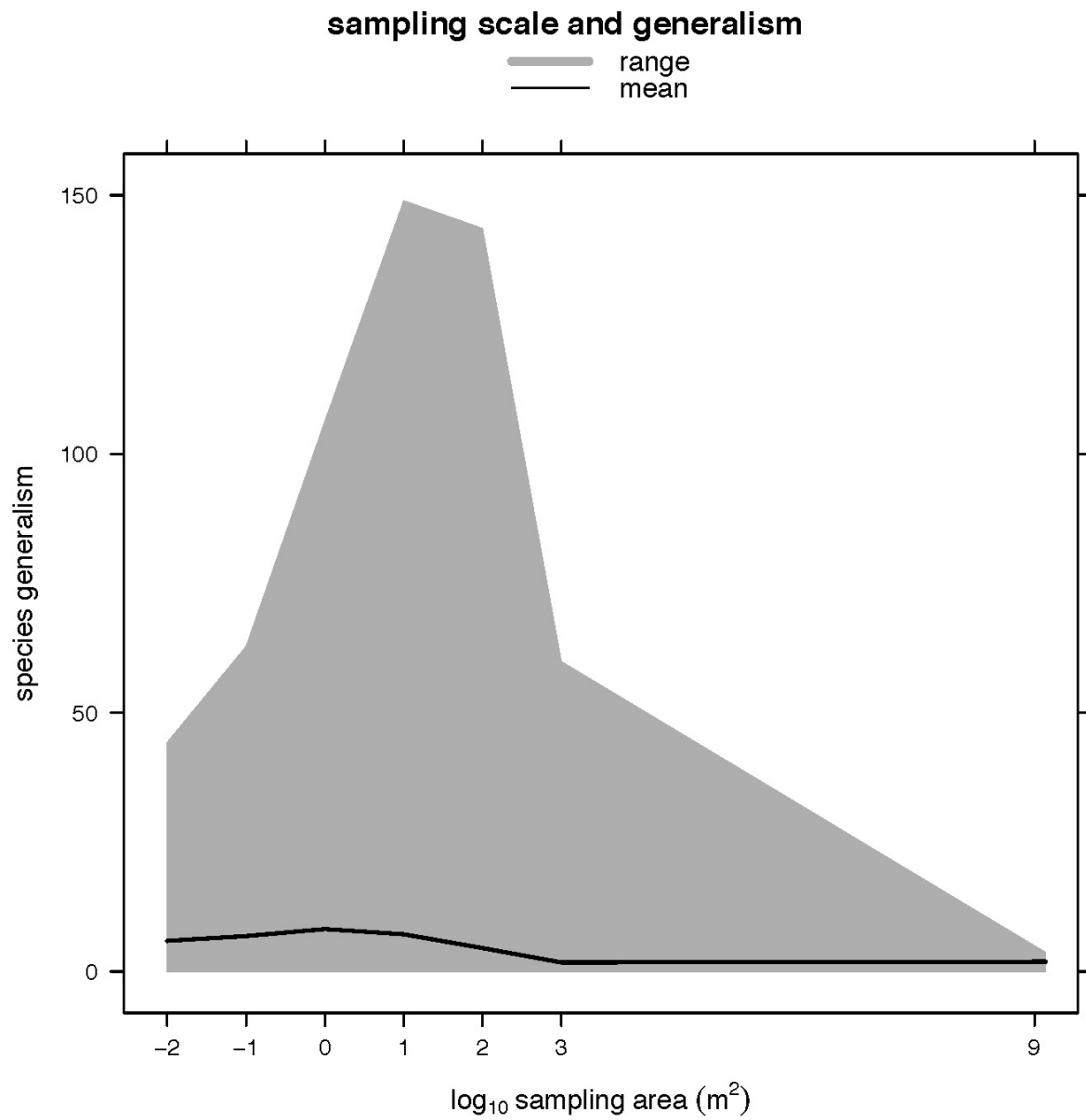


Fig. 2-8. Grain correction techniques. Mean variation in per-species width estimates is shown on the ordinate, at different levels of landscape homogeneity.

Analysis of Carolina Vegetation Survey data

The unmodified niche width estimates for plants in CVS tended to show a strong unimodal relationship with sampling grain, with maximum values at the 3.14 m² scale

(Fig. 2-9a). This suggests that Beta diversity is high at this scale, perhaps from abiotic heterogeneity or because at lower scales there is an element of pseudo-replication (modules from the same plot are close to one another). Even if not artifactual, this trend obscures differences between species. When values were normalized using a cumulative density function, a unimodal pattern persisted, though it was less severe (Fig. 2-9b). The narrowest of niches are observed at the smallest scale, but many species were most restricted at larger scales (Fig. 2-10); a plurality (33%) had their narrowest niche estimates emerge at the county scale. Generally, species showed a wide variety of responses to grain change (for example, consider those shown in Fig. 2-11).



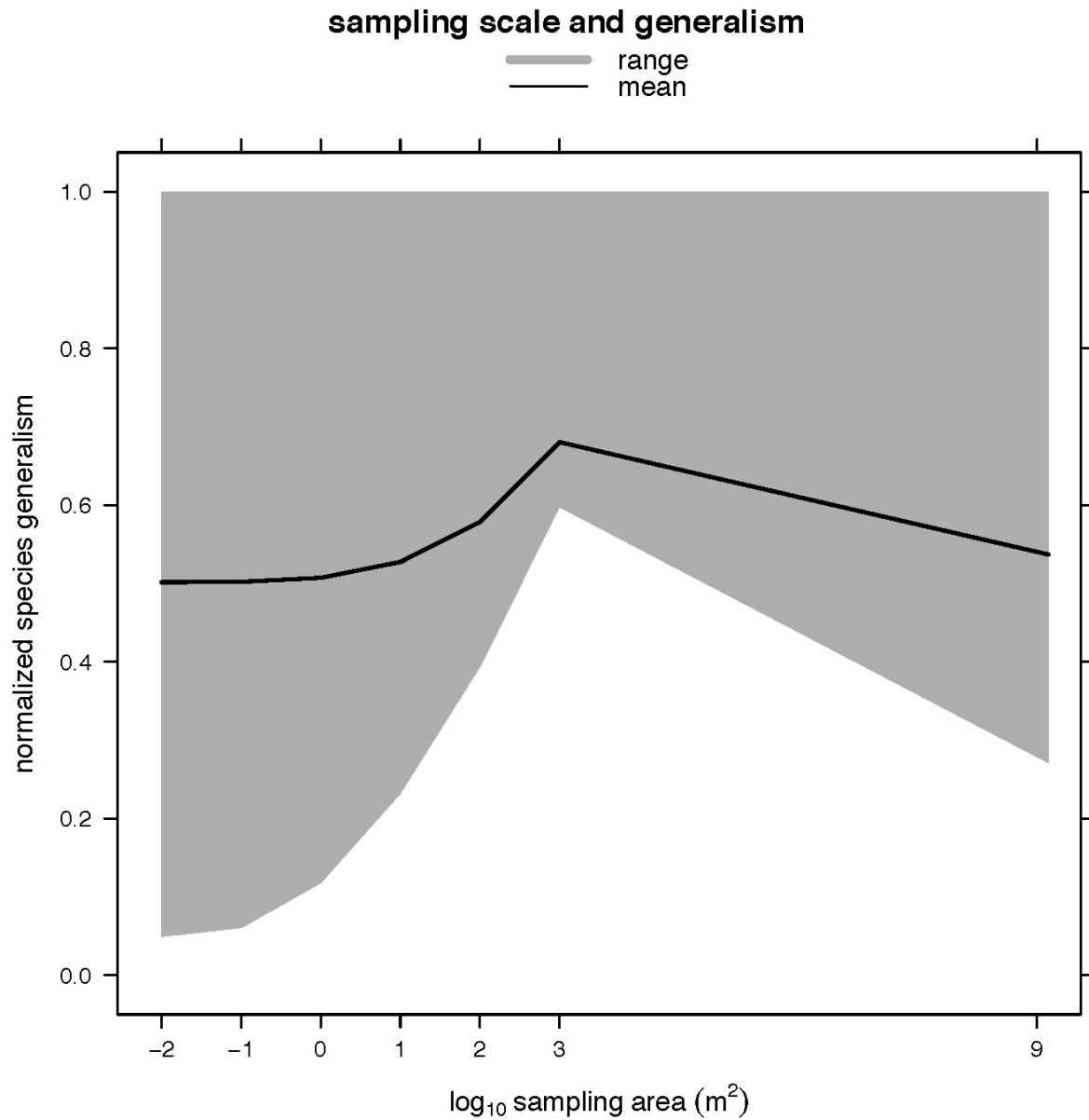


Fig. 2-9. Generalism scores for all sampled species in the Carolina Vegetation Survey data. Above: raw scores; below: scores normalized by cumulative density functions.

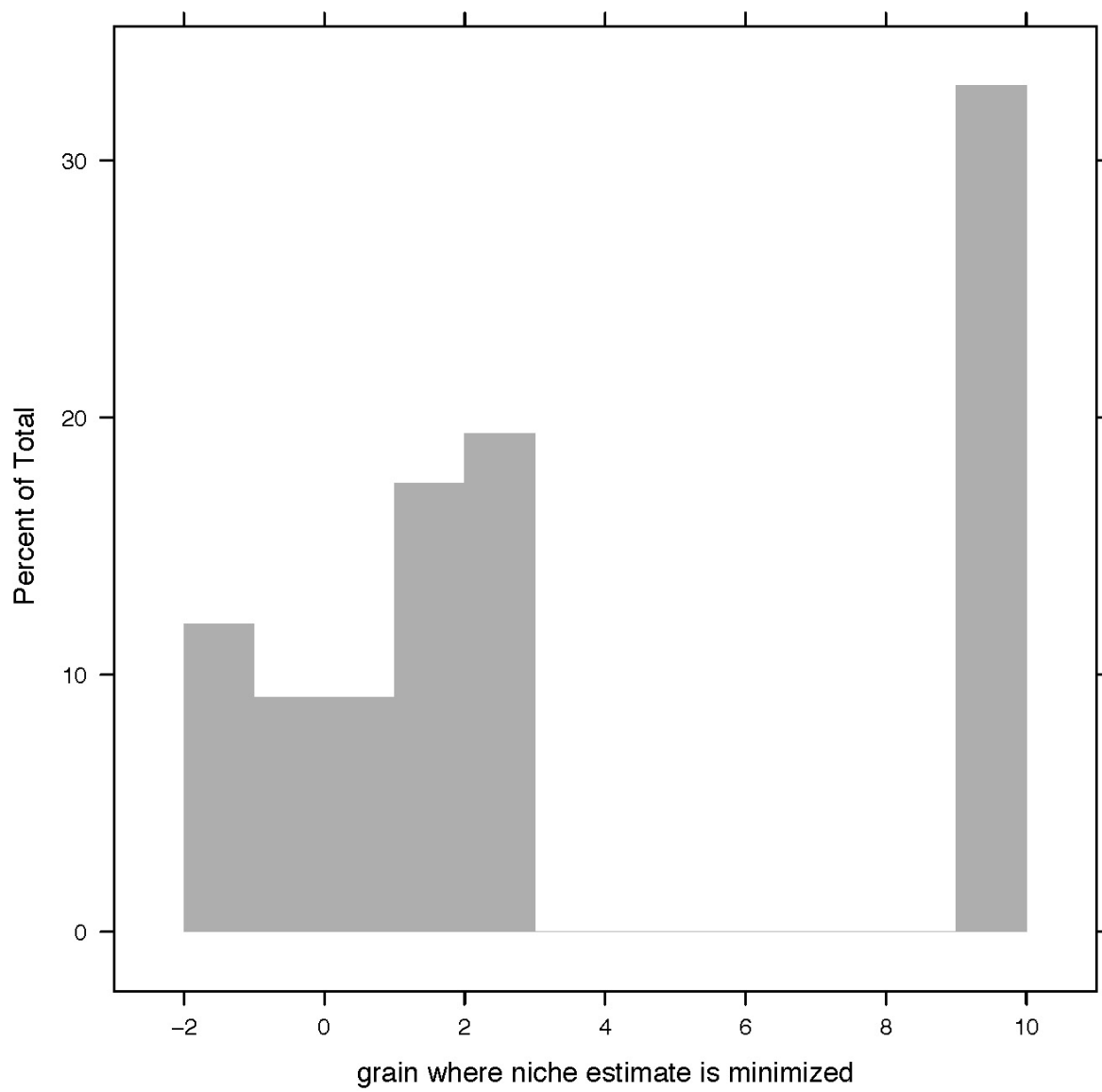


Fig. 2-10. Distribution of species' most limiting scale.

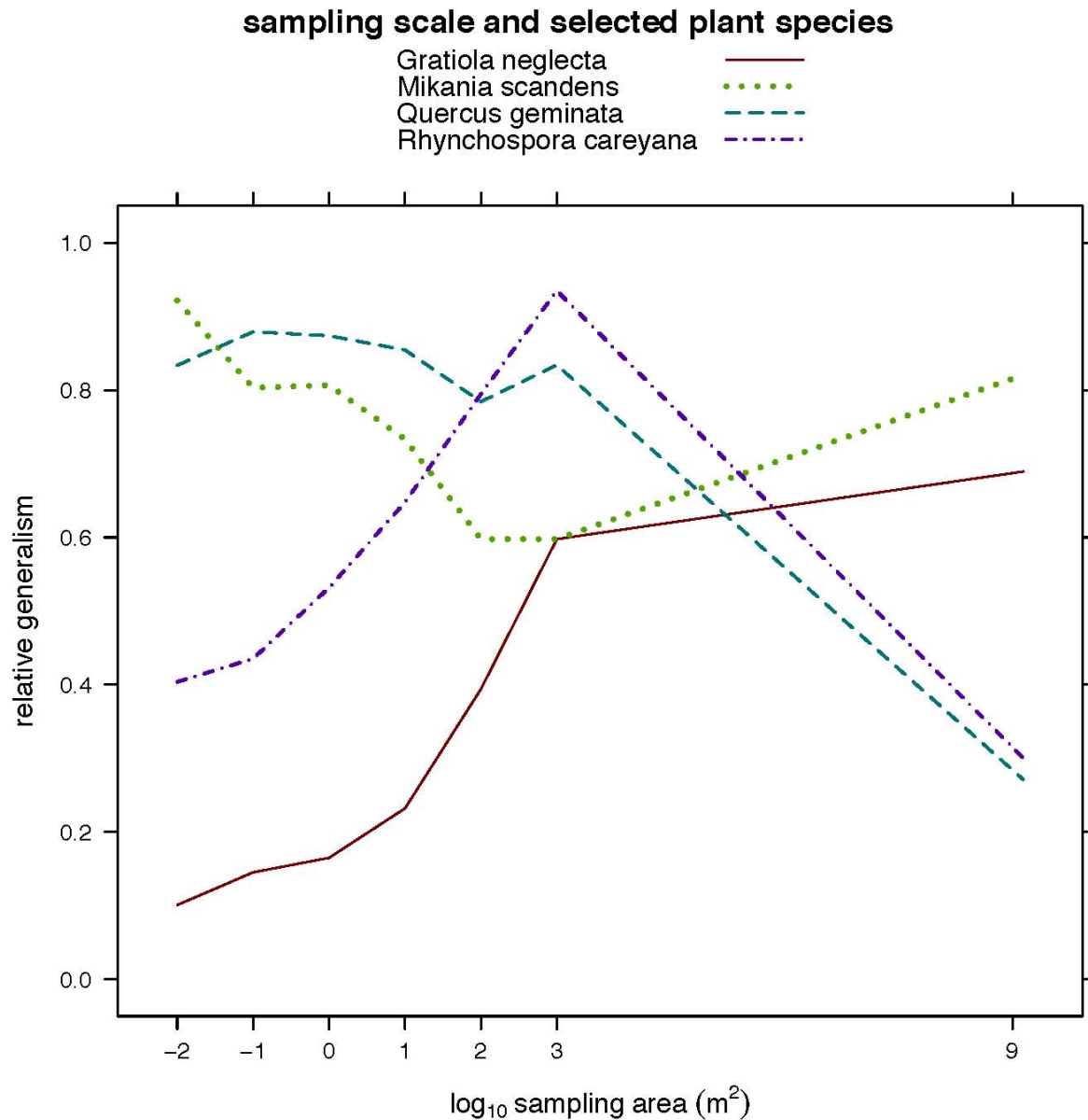


Fig. 2-11. Selected species illustrating different relationships between sampling grain and estimated niche width.

Growth habit and native status both affected the niche-grain relationship, with three noteworthy trends. Those plants that grow exclusively as trees were very general at small scales (below 1000 m²). The scores of generally smaller species (herbs and grasses) were much like those of trees at 1000 m² and remained high at the county-scale,

whereas trees and shrubs showed a decrease there. Non-native species exhibited higher generalism overall, but most especially at small scales (Fig. 12).

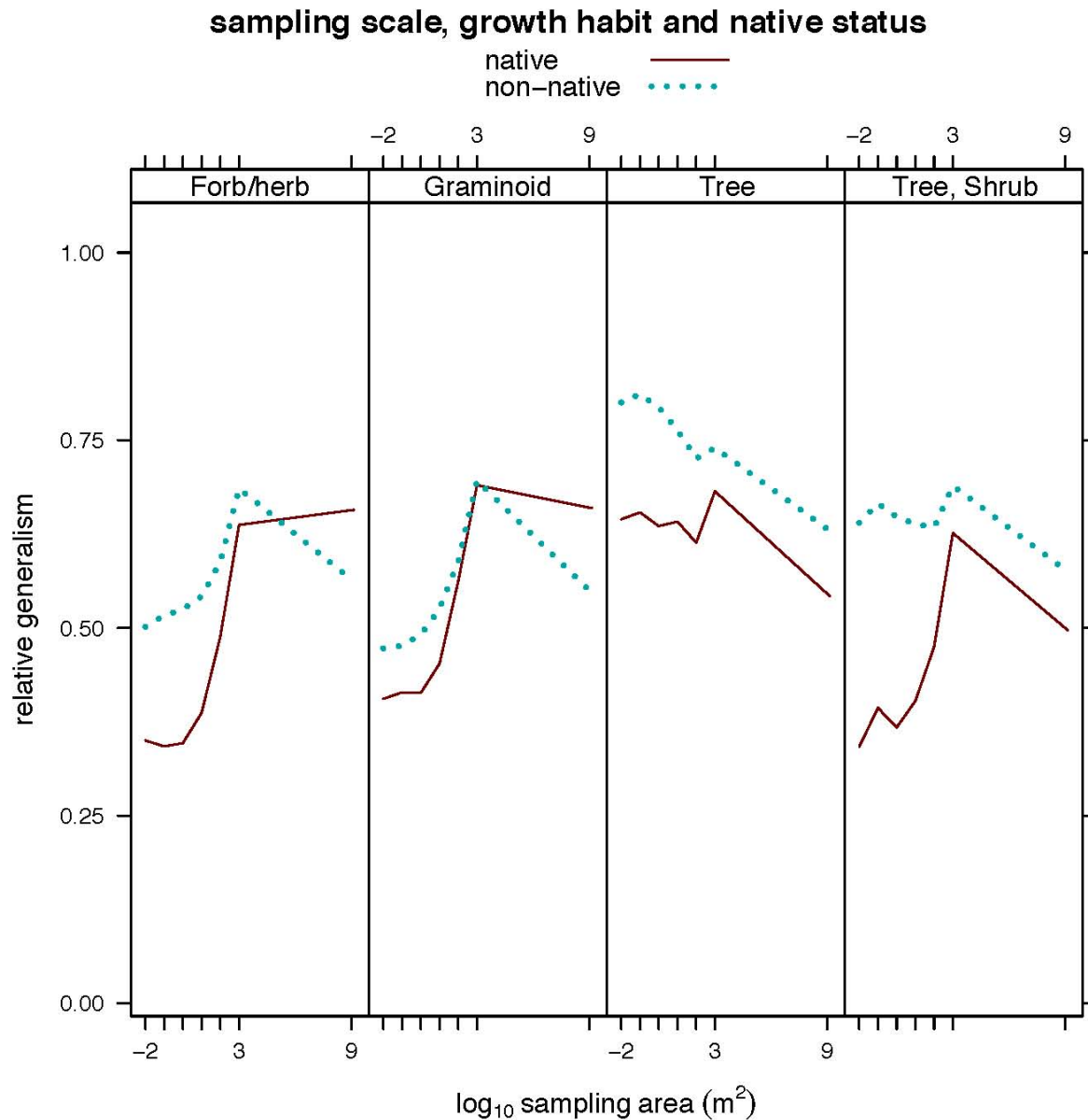


Fig. 2-12 Differences in generalism across grains by growth habit and native-status. Only the four most common growth habits are shown. Species were classified as non-native if invasive in any state, according to USDA PLANTS.

Discussion

Metric function

Metric performance is affected most drastically by in-plot resource heterogeneity. As this increases, increasingly dissimilar species will be counted as co-occurrences; and the difference between specialists and generalists will fade. In-plot heterogeneity is the product of heterogeneity in critical resources inherent in the landscape, and sampling grain. Metric performance in simulations was best when plots typically contained less than 10% of possible resource variation.

Good niche estimates cannot be achieved simply by minimizing grain: if it becomes too small, plots will capture no turn-over in the landscape and consist of monocultures, thereby eliminating all co-occurrence. Other problems are encountered above this extreme because a small grain likely implies sampling less overall space: as the sampled space decreases, co-occurrences are missed. This occurs with respect to the scale of turn-over in the landscape; when resources vary over a large scale, grain should also be large. (Sites could be chosen non-randomly to capture important co-occurrences, but this is also problematic.)

Thus, there is a balance between a tight definition of co-occurrence and thorough sampling. Unfortunately, it is difficult to know where this point lies in the real world, especially as multiple resources interact. Before a major study is initiated, a pilot study can examine many potential grains; those that maximize the variation in niche estimates should be preferred. As shown here, no single grain can adequately describe species niches except in extremely simple systems. Thus, the ideal remains multi-scale

sampling, as in the Carolina Vegetation Survey protocol (Peet et al., 1998), where the effect of grain can be analyzed, both in niche estimates and other ecological patterns such as species-area relationships.

Comparability

Niche estimates at different scales are not immediately comparable. This is because the beta-diversity of the landscape is likely to also vary with scale, and thus affect all species' estimates (as one interpretation of the metric is a modified, per-species, additive beta diversity measure). However, they may be compared when the scores at each grain are normalized by a cumulative density function.

Plant niches across scales

Plants recorded in the Carolina Vegetation Survey showed the greatest variation in raw generalism scores at level 2 sampling, i.e. 10 m². We may theorize that this scale, where mean richness is 6.3, gives the most accurate niche estimates overall (i.e. for all species, in all locations); which is useful to quickly summarize species' niches.

If we do consider species generalism across scales, though, there is a great deal of variety, as predicted. Many species were most limited at the large scale (i.e., by patterns and processes creating heterogeneity at the county (1.394×10⁹ m²) level), presumably reflecting broad ranges of soil type and climate (elevation and latitude).

Growth habit

Different growth habits and native status were associated with different responses. The niches of grasses and herbs were narrower than trees' at scales below 1000 m², but similar to or broader than trees' above that scale. Fine-grain limitation can be attributed to susceptibility to competition or the effects of micro-topography. Such factors will be important to species with smaller growth forms, like herbs and many grasses, that can be out-competed by shading, but unimportant to most trees, whose investment in physical size allows them to ignore micro-habitat except as seedlings.

The greater generalism of grasses and herbs at large grains may also owe to life histories strategies that allocate more energy to reproduction than those of trees or shrubs. A pioneer grass, for instance, can colonize and become established in suitable habitat even if the patch is disjoint from existing populations or very small; in either case, bringing the pioneer into close proximity with a wide variety of other species. Dispersal ability, which may be important to this explanation, can be seen separately from niche breadth, but a trade-off between the two is often theorized. Although the CVS database is biased towards mature communities, small-scale disturbance and heterogeneity within them may nonetheless show these characteristics of r-selected plants.

For specific examples, we can return to those illustrated in Fig. 11, chosen for their very different responses to grain. *Gratiola neglecta* (clammy hedgehyssop) is a small annual found throughout the US and Canada around muddy pond edges. Appropriate habitat such as that can be found commonly, but occurs only in narrow bands, limiting the

other species that can be close by. As might be expected, the species shows a positive relationship between grain and generalism.

Compare this to *Mikania scandens* (climbing hempvine), a weedy vine found in moist areas in the eastern US and Canada, with wind- and animal-born seeds. Sufficiently moist habitat is likely available in much larger patches than pond edges, so *M. scandens* is logically limited at a larger grain (10–100 m²) than *G. neglecta*, presumably the scale of turn-over for such patches.

Rhynchospora careyana (broadfruit horned beaksedge), in contrast, showed a strongly unimodal response to grain, with high generalism at the 1000 m² scale. It is found around limesink ponds in the southeast US and is strongly clonal. It may then be limited by competition or heterogeneity at very small scales, and the patchy availability of limesinks at large ones.

Finally, *Quercus geminata* (sand live oak) shows the pattern common to most trees: a negative generalism-grain relationship. *Q. geminata* is a shrubby tree that grows on xeric sites in the Southeast. Large propagules and strong clonal tendency likely allow establishment in many habitats at the small scale, while it is limited by climate (through light availability) at larger ones.

Native status

It is not surprising that non-native species generally have broader niches at all scales; if they had very specific requirements, they likely would not have become established.

This pattern is broken at the large scale for herbs and grasses though, where non-natives have narrower niches than natives. The implication is that counties vary widely in their suitability for non-natives; but differences within them are comparatively minor. This could be an artifact of recent colonization history: chance or perhaps human disturbance have introduced invasives in some areas but not others; within them, propagule pressure swamps all patches of suitable habitat. It may also suggest that non-natives are affected more by regional climate than natives.

Implications for ecological research

What does it mean to discuss a species's generalism changing with scale? The approach here has been to treat generalism as a shorthand for niche width, where "niche" subsumes the abiotic and biotic requirements of a species; i.e. it's tolerance of various conditions. In this framework, a species has but one true niche but many realized niches. Ecologists have long known that any measurement of niche is of the latter kind, showing but a piece of the whole, determined by limited methods. If we allow "niche" to include a species' responses to local disturbance and micro-habitat, in addition to broad climatic variables, grain must matter as much in sampling as does extent, for just as we would ideally sample across the temperature regime of a species' distribution, so should we sample across grains to detect the variety of processes limiting the species' appearance.

Most distributions are patchy, and patchy at multiple scales, which suggests that studying apparent niche across grains may be important for understanding those distributions. Such an approach may also be relevant for studying different aspects of a

large process, including those relevant for management. For instance, in exotic invasions, we might need the right grain to detect signals of dangerous niche expansion in a potential invader, or any unoccupied niche “holes” in communities – suggesting invasability. The use of generalism estimates based on co-occurrence, when employed across scales, should thus be a useful tool in the ecologist’s belt.

Works Cited

- Levin, S. A. (1992) The Problem of Pattern and Scale in Ecology: The Robert H. MacArthur Award Lecture. *Ecology*, **73**, 1943–1967.
- Palmer, M. W. & White, P. S. (1994) Scale Dependence and the Species-Area Relationship. *The American Naturalist*, **144**, 717–740.
- Wiens, J. A. (1989) Spatial Scaling in Ecology. *Functional Ecology*, **3**,
- Højsgaard, S. (2009) doBy: Groupwise computations of summary statistics and other utilities. R package version 3.7.
- Hurlbert, A. H. & Jetz, W. (2007) Species richness, hotspots, and the scale dependence of range maps in ecology and conservation. *PNAS*, **104**, 13384–13389.
- Fridley, J. D., Vandermaast, D. B., Kuppinger, D. M., Manthey, M., & Peet, R. K. (2007) Co-occurrence based assessment of habitat generalists and specialists: a new approach for the measurement of niche width. *Journal of Ecology*, **95**, 707–722.
- Nekola, J. C. & White, P. S. (1999) Special Paper: The Distance Decay of Similarity in Biogeography and Ecology. *Journal of Biogeography*, **26**, 867–878.
- Peet, R. K., Wentworth, T. R., & White, P. S. (1998) A Flexible, Multipurpose Method for Recording Vegetation Composition and Structure. *Castanea*, **63**, 262–274.
- Peet, R. K., Wentworth, T. R., Schafale, M. P., & Weakley, A. S. (2004) Carolina Vegetation Survey database. Version 3.0. North Carolina Botanical Garden. Chapel Hill, NC 27599.
- R Development Core Team (2009) R: A Language and Environment for Statistical Computing. Version 2.8.1.
- Reed, R. A., Peet, R. K., Palmer, M. W., & White, P. S. (1993) Scale Dependence of Vegetation-Environment Correlations: A Case Study of a North Carolina Piedmont Woodland. *Journal of Vegetation Science*, **4**, 329–340.

Sarkar, D. (2009) lattice: Lattice Graphics. R package version 0.17-20.

Schlather, M. (2009) RandomFields: Simulation and Analysis of Random Fields. R package version 1.3.37.

USDA, NRCS. (2009) The PLANTS Database (<http://plants.usda.gov>, 18 September 2009). National Plant Data Center, Baton Rouge, LA 70874-4490 USA.

Wickham & Hadley (2007) Reshaping data with the reshape package. *Journal of Statistical Software*, **21**.

3. A Wealth of Niches

Detecting community processes with species generalism

INTRODUCTION

Ecologists are interested in the processes forming communities, and the variation within them, such as the relative importance of the regional species pool and local filters such as competition and disturbance. But these are difficult to observe directly. Experiments showing how a community responds are expensive and cannot be performed everywhere, much less on rare communities. Thus, we would like methods to infer process from data.

Pattern begets process and process begets pattern. If a process is important, it necessarily leaves a signature, and this should be identifiable. We should look for signatures in community composition that are easily measured, universally present across systems, and variable. Richness obviously fits these criteria but is not sufficient because too many events can lead to it. Diversity exists in many forms though: richness not just in species but trophic positions, traits and strategies (Diaz, 2001).

A simple definition of strategy is generalism, or niche width, whereby some species have many requirements and others few (Colwell & Futayma, 1971). This idea is broad enough that it applies to all species, and values are easily estimated (Fridley et

al., 2007, McChesney ms). Generalists and specialists should respond very differently to some important processes, so that some communities might have all specialists, all generalists, or some other distribution.

In this chapter I ask whether the importance of several broad types of process can be seen in the patterns of generalism and richness within sites. I first assess this with simulation modeling, where known processes are used to generate patterns. I then consider real data, looking for equivalent patterns and comparing them to what is known about those communities.

FOUR SIGNATURES

Based on the strength of several processes, we can expect distinct relationships between in-plot richness and generalism—or its estimate based on co-occurrence data, θ (Fridley et al., 2007). Distributions of θ can be described by its mean, standard deviation, maximum and minimum per plot. For simplicity, I consider four signatures, generated by a single process each—and named for generalism’s role in that process—but intermediate types are possible.

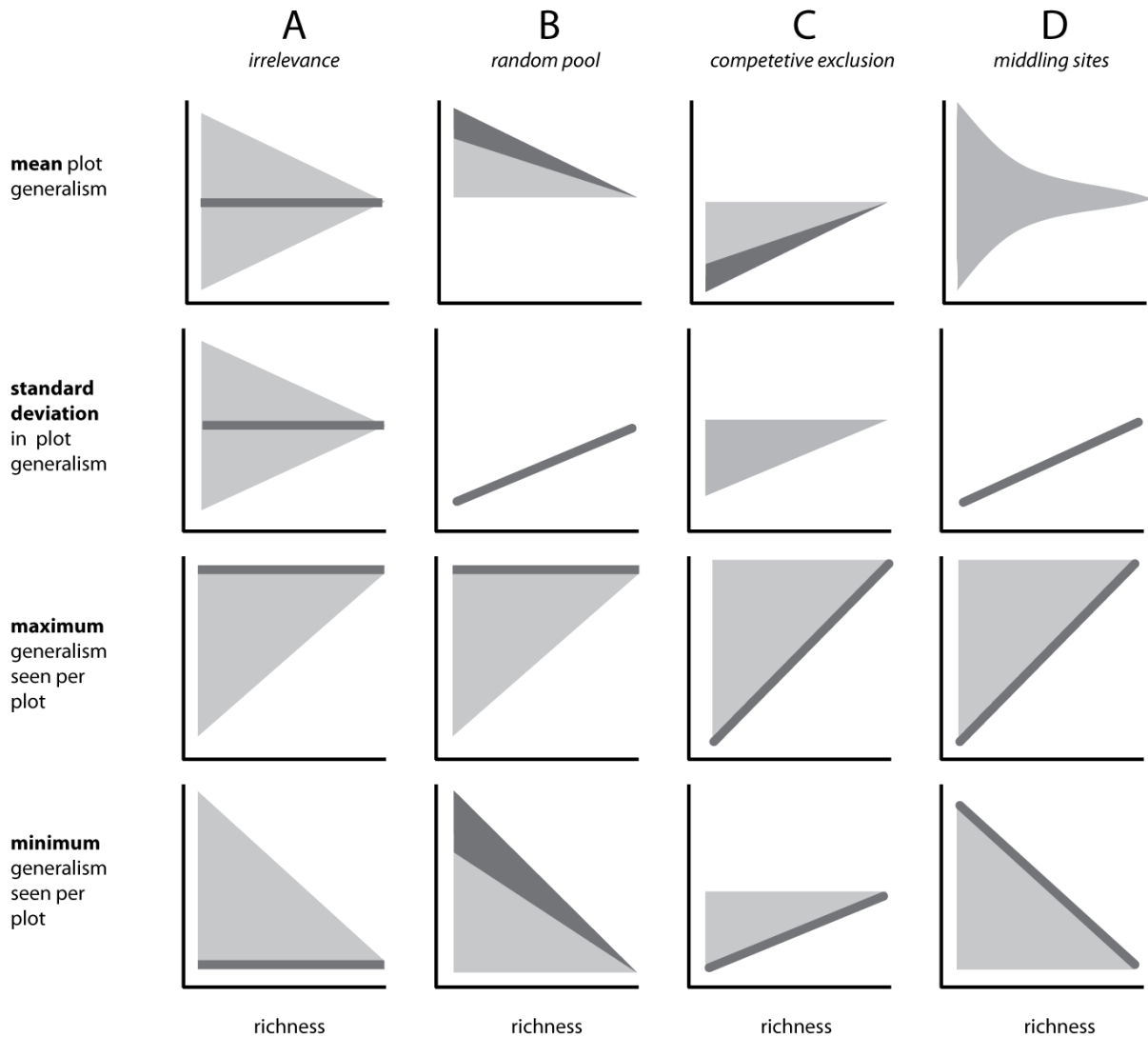


Fig. 3-1. Hypothesized richness-generalism patterns in four pure models. The generalism scores of all species in a plot are summarized by mean, standard deviation, minimum and maximum. Dark gray shows the over-riding trend while light gray shows the range of possible values. (A) is the null model, where generalism and richness are decoupled except for statistical constraints at very high richnesses. In (B) richness depends on species availability; generalists are always available so dominate poor sites. In (C) we assume a competition-generalism trade-off, so that where specialists happen to occur, generalists do not. In (D) richness is generated only when generalists and specialists co-exist, while poor sites favor one strategy.

A. Irrelevance

Our null model is that generalism is unrelated to site richness. Some sites are inherently rich, and some poor, due to unrelated factors like heterogeneity or resource abundance (Tilman, 1994). The lack of correlation may result from realized niches that are very different from intrinsic ones; meaning that estimates are poor. This could result from poor data or the importance of historical contingency in the study community.

Even under this model, some patterns should be expected when graphing summary statistics of generalism against richness. Sites with moderate richness can have any composition of generalists or specialists. But highly rich sites must necessarily contain a large proportion of the local pool, and thus its mean generalism is constrained to match that of the available species. Deviation in generalism is unaffected by richness, while minima are uniformly low and maxima high (Fig. 3-1A).

B. Random pool

Even if generalism is not directly involved in community formation, as in (A), pattern might emerge because of differential availability of specialists and generalists. In other words, local filters may be much less important than stochastic, regional processes (Ricklefs, 1987).

A species is available to colonize a site only if its niche intersects the qualities of the site in n -dimensional niche space; i.e., if its requirements are satisfied. Let us assume

species do not interact (as they will in later models), that colonization is entirely random, and species are distributed evenly through niche space. Almost all sites can be colonized by some generalists, which by definition cover wide swathes of niche space. But not all sites will happen to suit specialists—some will suit a few, others none.

Therefore, the difference between rich and poor sites will hinge on the presence of specialists. The richest sites will necessarily have both (with intermediate mean theta and high deviation), while the poorest sites will have only generalists (low mean and low deviation). See Fig. 3-1B.

C. Competitive exclusion

Evolutionary trade-offs may be important in community assembly (Bonsall, 2004) and one possible trade-off pairs generalism with competitive ability. This has been predicted by allocation theory (Levins, 1968; Gilchrist, 1995) and had some empirical support (e.g., Dykhuizen, 1983). If true, a niche cannot be both wide and high—where height represents resource acquisition rate (Tilman, 1982).

In this model, we posit such a trade-off and allow interactions between species. Where resources are scarce, specialists competitively exclude rivals. Where resources are abundant, competition is less important and generalists win through rapid colonization (MacArthur and Wilson, 1967). We can easily imagine such sites in a system driven by disturbance, or where only a few specialists are present.

In this model, specialists will tend to occur alone, and only sites that lack specialists (due to evolutionary or ecological chance, or disturbance) can attain high richness. Thus, all aspects of generalism should be correlated with richness, especially maxima: poor sites should not contain many generalists at all (Fig. 3-1C).

D. Middling qualities

Several theories posit increased richness at sites with middling attributes, e.g., intermediate disturbance or intermediate resource quality (Connel, 1978; Peet and Christensen, 1988). One common idea is that sites may favor different classes of species (e.g., high disturbance favors generalists whereas low disturbance favors specialists). Non-favored classes are excluded, lowering richness; only the sites that are intermediate, and so exclude nothing, can attain high richness.

Here we expect rich sites to contain both specialists and generalists, and so have intermediate mean theta, and high deviation. Poor sites can show a wide variety of mean values but always low deviation. A narrowing mean is also expected from simple statistical effects, as described for (A). Thus, here we actually expect an exaggeration of that trend as the two causes should be multiplicative; poor sites should vary greatly, but even moderately rich sites have highly constrained mean thetas.

METHODS

Simulation Models

Using a previously built simulation model (McChesney, in draft) I generated plot data for several exemplary scenarios. In each simulation species niche curves were placed on three independent resources axes. Values for each resource were randomly assigned to each plot, limiting the potential colonizers to species tolerant of those conditions. From this local pool, species were randomly drawn.

Most simulation parameters were fixed (Table 3-1) while three were modified to generate the four pure scenarios (Table 3-2). Plot richness could be controlled directly with the number of species draws. In scenario A, this was randomly varied between plots and all species were available for drawing, irrespective of niche. A parameter for inter-specific interaction strength determined the importance of niche curve height in the draw process. A trade-off flag indicated whether niche height was inversely correlated with width.

Plots were summarized by richness and the generalism of all species present. Each scenario was replicated 10 times. In order to compare results, both richness and generalism were scaled per replicate, to vary between 0 and 1.

Table 3-1. Fixed simulation parameters. In the scenarios testing trade-offs, niche optima heights (competitive ability) were not generated but taken as the inverse of breadth (generalism).

parameter	function	value
n species	constant	50
niche breadth	gamma	mean=1.0
optimum position	uniform	range=0-1
optimum height	lognormal	mean=2, SD=1
plots	constant	500
spp. draws per plot	constant	100

Table 3-2. Parameters varying by scenario. Species draws controls site-specific, abiotic richness. Interaction strength controls the force of competition or other exclusionary forces, where low optima lose to high optima when the value is positive; in scenario D sites were randomly assigned values which could be negative, thus favoring low optima. The trade-off flag controls whether optima height are randomly generated or based on niche breadth.

scenario	A	B	C	D
species draws	10–100	100	100	100
interaction strength	0	0	3	-3–3
trade-off	no	no	yes	yes

Carolina Vegetation Survey Data

Data from the Carolina Vegetation Survey were then analyzed (Peet and Christensen). All species in 1000 m² plots, entered by May 2009, were considered: this came to 3880 species in 2804 plots.

Relative abundance was computed from estimated cover values (the minimum and maximum values defining each cover class were converted to a single value by finding their geometric mean). The resulting abundance matrix was used to estimate

species niches with the generalism metric put forward previously (McChensey, in draft), based on the Fridley et al. (2007) theta metric. Richness and summaries of generalism were then computed per plot.

Other Complex Systems

Anthropologists have occasionally attempted to use niche theory in human-ecological systems (Hardesty 1975), and other work has shown that patterns seen in ecological systems, like species-area curves, can appear in a wide variety of non-biological complex systems where discrete items appear in groups (Nekola, 2007). Using the methods described above, brief analyses were performed on data from two human systems: one agro-ecological, one wholly symbolic.

In 1066 the Normans conquered England and rapidly undertook an assessment of its wealth in the Domesday Book (Palmer et al. 2007). This includes counts of seven major livestock types at each manor. Presumably, human interests constrain the species pool and minimize competition and other constraints; but perhaps more general concepts, like the exclusion of similar species, also serve those interests, creating similar patterns. These data were analyzed, with manors counted as locations.

Any written text shares properties with an ecological system: words can be seen as species that appear in various combinations in sentences of various length (richness). One might be inclined to abandon the search for particularly ecological processes if very similar patterns were obtained from the text of Charles Darwin's

“Variation of Plants and Animals under Domestication” (1868), which were also analyzed.

Matching

Generalism-richness relationships were summarized by the slopes of linear models, with normalized richness. Each model or system was described by four slopes representing the four parameters shown in Figure 3-1: mean, standard deviation, maximum and minimum. The expected patterns are qualitative though, so comparisons between systems were made using the signs of the slopes: positive, negative or zero. Very small slopes ($<3\%$) were treated as zero. A pattern was considered to match another when the signs of all four relationships were the same. Where multiple matches were possible, the null model (A) and near-null model (B) were preferred.

RESULTS

Simulation models

When the four pure models were simulated, the patterns of generalism and richness (Fig. 3-2) qualitatively matched what was theorized (Fig. 3-1). The main differences were more diffuse relationships, and only a mild response from standard deviation to richness.

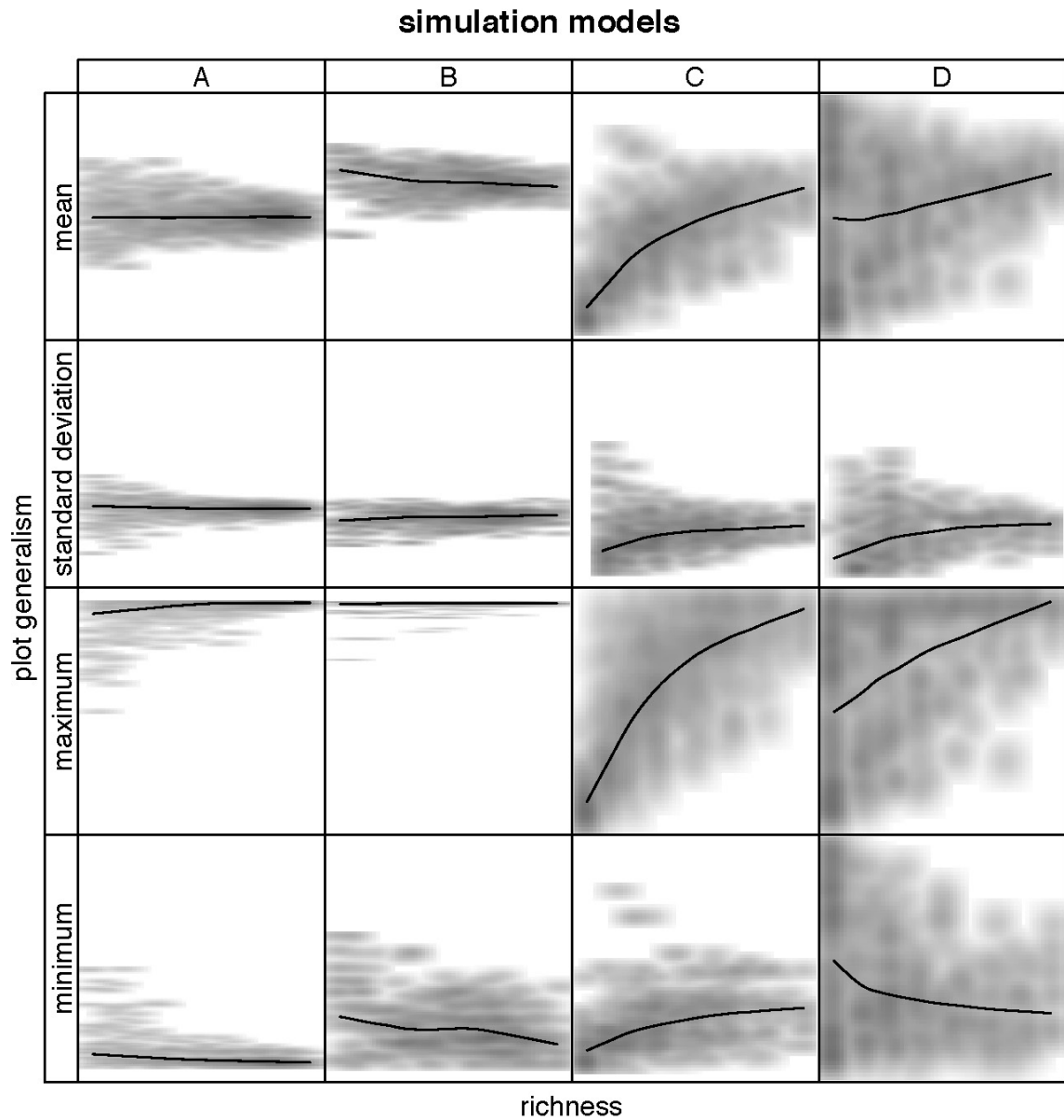


Fig. 3-2. Simulation model results for four pure scenarios, A–D, where each shows 10 replicates overlaid. Some relationships are obscured because replicates had different levels of richness.

Table 3-3. Relationships between four per-plot generalism statistics and plot richness under the four models. Slopes were calculated by linear model for each of 10 replicates, and the values averaged, with non-significant slopes counted as 0.

model	A	B	C	D
mean	0.00	-0.04	0.57	0.23
S.D.	0.00	0.02	0.09	0.05
min.	-0.09	-0.13	0.21	-0.10
max.	0.09	0.03	0.95	0.56

Carolina Vegetation Survey data

Considering all plots in CVS together, generalism patterns resembled those of models B and D. Plots showed wide variation in the generalism of their species when richness was low, but when it increased, the mean, standard deviation and minimum became constrained (Fig. 3-1). Linear models showed that for each of the four summary statistics, 1% to 8% of variation could be explained by richness (Table 3-4; all p values < 0.001). Mean generalism decreased while minimum and maximum became more extreme and standard deviation declined slightly.

CVS plot diversity and generalism

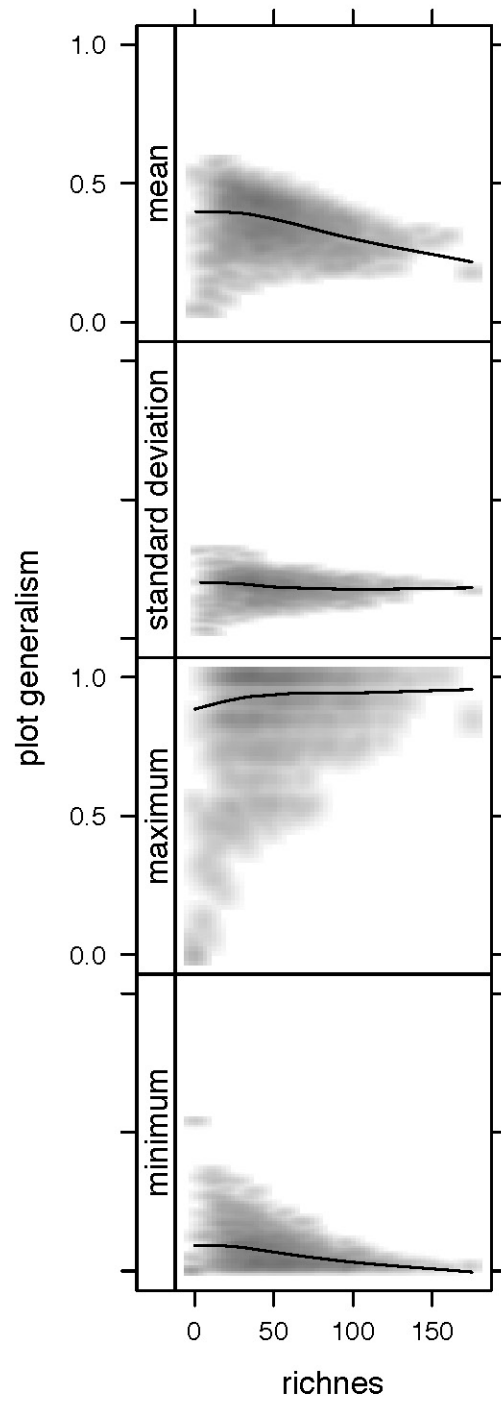


Fig. 3-3. Generalism and richness patterns in plants of the Carolina Vegetation Survey.

Table 3-4. Dependence of plot generalism on richness in all of CVS. Linear models relating each plot-level generalism statistic to richness.

statistic	r²	slope	p
mean	0.08	-0.18	2.2e-16
S.D.	0.01	-0.03	5.3e-6
maximum	0.04	0.29	2.2e-16
minimum	0.15	-0.15	2.2e-16

Taken as a whole, plots in CVS do not strongly match any of the four models: no model has a negative slope for the mean or standard deviation of generalism.

Variation between community types

The patterns seen across CVS were not replicated in every community type. Linear models showed that within most community types, there was no significant relationship between generalism and richness (Fig. 3-4); though when there was, it was almost always negative, for example in the six most common communities (Fig. 3-5).

generalism within CVS projects

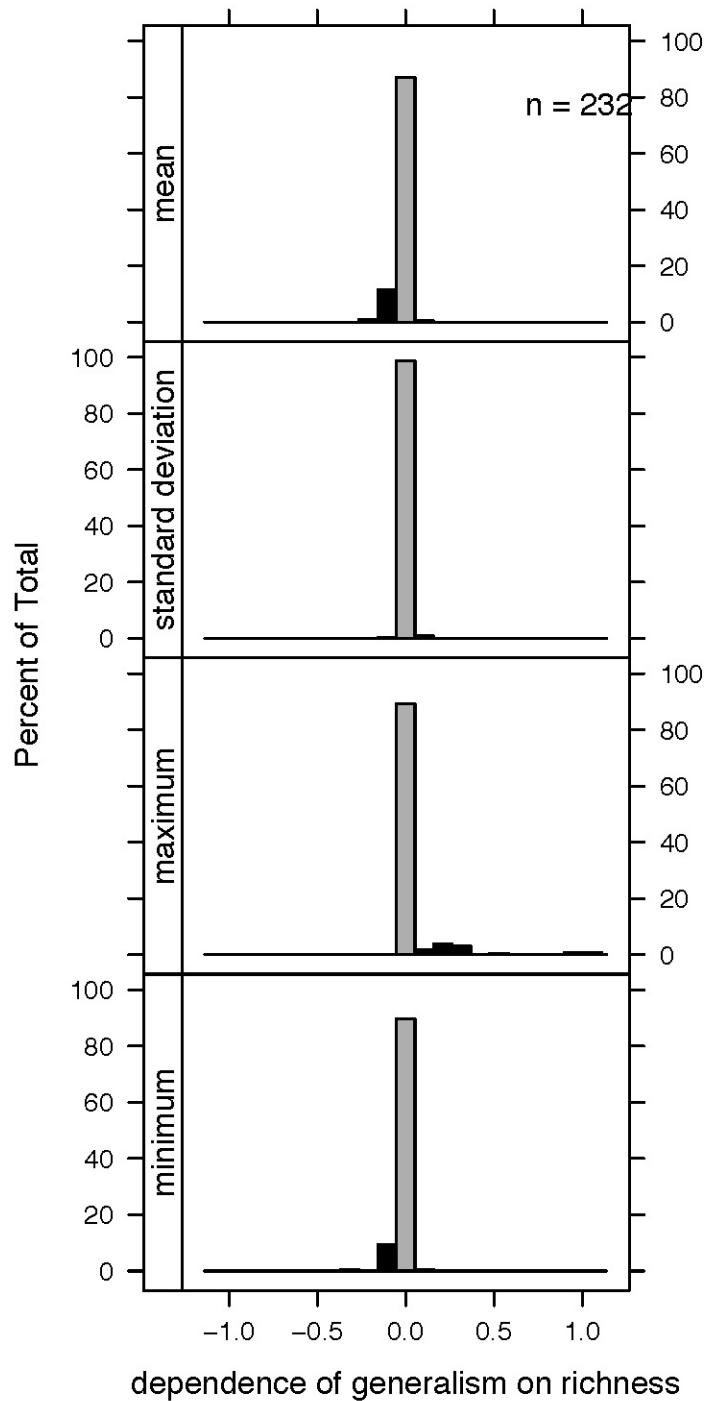


Fig. 3-4. Generalism-richness relationships in CVS communities. Each datum is the slope of a linear model that compares plot generalism to richness in one community type. Communities appearing less than three times were excluded. The histograms show different statistical summaries of plot generalism.

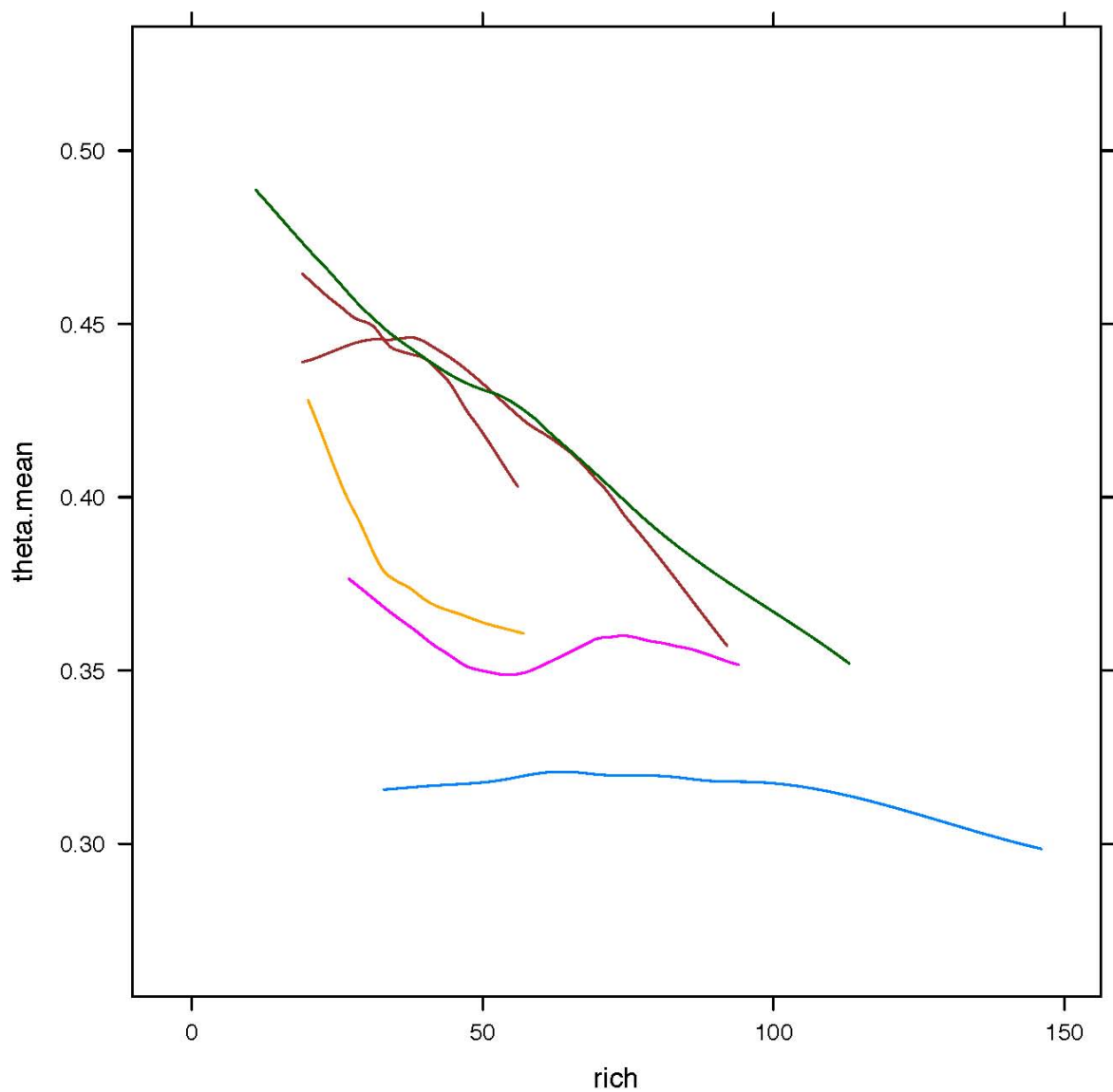


Fig. 3-5. Loess-smoothed relationship between plot richness and mean plot generalism in the six most common community types in CVS.

The qualitative generalism-richness relationships in each CVS community type were compared with those of the four models. Of 232 recorded community types, 127 were not represented sufficiently for slope estimation, leaving 105 that were

matched (see Appendix B). Communities were most often matched to model A, while many matched no model (Table 3-5).

Consider the “Southern Appalachian Cove Forest (Rich Montane Type)” community (CEGL007695) which is dominated by *Aesculus flava* and *Acer saccharum*. Plots in CVS had richnesses between 33 and 146, with a mean near 72. The richer sites had more extreme generalists and specialists, which is expected for all models except C; they also showed greater mean generalism and greater variation in generalism, which is expected only under model D.

Table 3-5. Communities in CVS matching each of the proposed models in richness-generalism relationships. A match was defined as four slopes (between richness and mean, minimum, maximum or standard deviation of generalism) with the correct sign: positive, zero, or negative. Slopes of low magnitude (<5%) were treated as zero. Where multiple matches existed, the null model (A), and then the simpler (B) were preferred.

model n matches		example (common name)
A	62	High-Elevation Red Oak Forest (Deciduous Shrub Type)
B	19	Appalachian Montane Oak Hickory Forest (Typic Acidic Type)
C	1	Calcareous hammocks & shell barrens
D	1	Southern Appalachian Cove Forest (Rich Montane Type)
none	22	Southern Appalachian Acid Cove Forest (Typic Type)

Other Complex Systems

The Domesday Book records seven major types of livestock on England’s manors in 1066. Cobs (Welsh warhorses) showed the highest generalism while goats showed the lowest; cobs appeared with many other animals and presumably on a wide variety of manors, while goats did not. The relationship between the richness and generalism of a manor’s stock approximated the patterns seen in model B (Fig. 3-

6A). The same pattern was seen even more strongly in the sentence-word relationship in the Darwin text, being very linear with little dispersion (Fig. 3-6B).

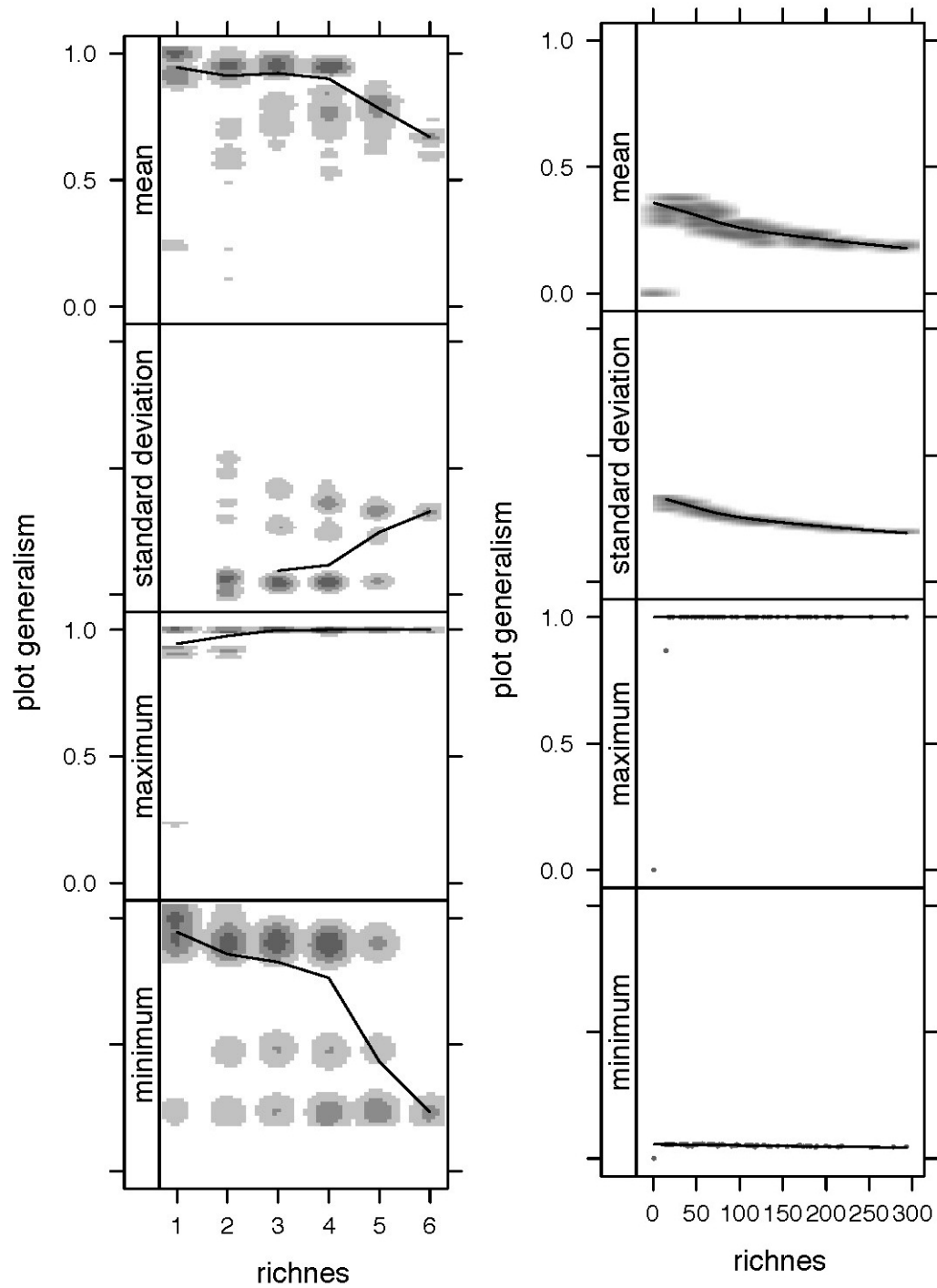


Figure 3-6. Richness and generalism relationship in two human systems; one agricultural, one symbolic. In (A) livestock presence on English manors in 1066, as recorded in the Domesday Book. In (B) words in Darwin's "The Variation of Animals and Plants under Domestication," where words are treated as species and sentences as sites.

DISCUSSION

Pure models

Patterns seen in the simulation models generally matched those expected from abstract ecological reasoning. Standard deviations showed less dependence on richness than expected, but nonetheless, the four models showed qualitative differences that can be sought elsewhere. C is readily identified with a positive minimum and mean. B has a negative mean (with a magnitude about as great as its minimum). A and D, in contrast, have little slope in mean or deviation. They can be told apart by the steep slopes of D's minimum and maximum; far greater than that of its mean.

Of course, these four signatures blend in to one another, as each suggests the power of certain processes that are not exclusive. The more a system's generalism pattern looks like C or D, the more we suspect that specialists are being favored when they appear; and, if like D, sometimes disfavored. The more like A it appears, the more we think that generalism has no importance (perhaps because it has been mis-measured).

Between-community patterns

Considering all plots in CVS simultaneously, a pattern reminiscent of models B and D emerges. This suggests that across a wide range of communities, generalism plays several roles in community formation.

Rich sites tend to contain more specialists than species-poor ones. This could be due simply to an uneven distribution of species in niche-space and thus the local pool: some communities happen to have specialists available to them but almost all have generalists. The “happenstance” of such availability is dependent on ecological processes that alter the regional pool, such as range expansions, and also evolutionary ones: if some habitat types are abundant, more species may have evolved there and experienced more intense niche partitioning. That generalism matters in this simple way is not surprising, but confirms that our estimates of niche width are not terrible and the suitedness of species to their habitat is not washed out by stochasticity (e.g., in dispersal).

To some extent, though, CVS plots also resemble model D, where extreme specialists or generalists are not seen together except at the very richest sites. This suggests that some plots favor specialists while others favor generalists. The mechanism behind this may be a trade-off making specialists competitively superior to generalists in their preferred habitat. Generalism is also often associated with good dispersal ability. So sites with high disturbance frequency should support generalists, while stable ones support specialists. Only intermediate sites, favoring neither strategy strongly, can support both.

In dividing sites into specialist-supporting or generalist-supporting, though, we are only describing a small effect. CVS mostly resembled model B, not D, and moderate generalists are found everywhere. Thus, trade-offs and competition are important but a small factor when compared to all the other forces operating across the wide range of sites sampled.

In-community patterns

In contrast to the above, when we look within CVS communities, there is generally no association between richness and generalism; not even via the effect of generalists appearing commonly. The processes creating such a pattern may be active but masked by other forces. This could occur if differences in generalism were sorted out at the community level, so that little variation appears within them.

Such low variation could originate in at least three places. (1) Our sample sizes of each type may simply be too small. (2) Our communities definitions may be sufficiently tight that large variation is ruled out by definition (implying that beta-diversity in non-dominant or non-indicator species is quite low). (3) Niche overlap may not be random but instead highly structured, e.g., niches could be arranged in n-dimensional space so that at any point there is one generalist and two specialists, even though the identity of these species may vary across a gradient. This could arise from various species-sorting processes (both ecological and evolutionary), which may help maintain identifiable communities. Such sorting would be expected within but not between communities, as that is where species interactions take place.

A subset of CVS communities did display similarities with model B, suggesting that at least the commonness of generalists mattered. These communities were primarily high-elevation hardwood forest types. Presumably these types contain more variation in generalism than many other southeastern communities; i.e. suffer none of the constraints on such variation enumerated above. This may make sense, as these community concepts are not highly restrictive or rare.

Human systems

Both systems appeared like model B, in which generalism matters in community formation but competition and trade-offs do not. Some parallels to biological generalism seem clear: in English manors, domesticated animals that are economically viable in a range of environmental and local social situations; and in Darwin's writing, linking words and those expressing general concepts rather than specific, technical ones. The less obvious parallel is to competition: some tendency for specialized animals or words to exclude others. The lack of such a force is also not surprising: farms can raise a large number of animals before overhead costs impose any limit, and even technical sentences require conjoining words. Non-biological systems that do display such "competition" are conceivable though, e.g., if the cost of maintaining different animals were much higher in some other agro-ecosystem, it might be observed. The methodology of considering the generalism of discreet items that appear in collections is thus applicable beyond ecology.

Conclusions

We can say some tentative things about the forces structuring plant communities in the southeastern United States. Species generalism does not seem to be involved in the small differences between sites of a basic type, except, to a limited extent, in some very common communities. But taking all communities together, we see slightly more specialized species in rich sites; and in poor ones, a tendency toward specialism or generalism but not both. Possibly, these is because specialists are competitively dominant, and exclude generalists, and also because sites with

intermediate qualities (disturbance, resource levels, etc.) support both specialists and generalists.

The analysis of generalism patterns in ecological communities can suggest the relative importance of major types of ecological process. For instance, in CVS, the community showing the greatest correspondence with model D was South Atlantic Coastal Plain Longleaf Flatwoods. We might hypothesize that competition and differences in local disturbance regime are unusually important in structuring this community; and we could proceed to test this empirically. It should also be possible to perform analyses across time to infer changes in process strength, e.g. after an invasion. Community generalism analysis is thus a potentially useful tool that can be applied across a wide variety of systems to address various ecological questions.

WORKS CITED

- Bonsall, M. B., Jansen, V. A., & Hassell, M. P. (2004) Life History Trade-Offs Assemble Ecological Guilds. *Science*, **306**, 111–114.
- Colwell, R. K. & Futuyma, D. J. (1971) On the Measurement of Niche Breadth and Overlap. *Ecology*, **52**, 567–576.
- Connell, J. H. (1978) Diversity in Tropical Rain Forests and Coral Reefs. *Science*, **199**, 1302–1310.
- Darwin, Charles (1896) Variation in Plants and Animals Under Domestication. *Project Gutenberg*, 2001.
- Díaz, S. & Cabido, M. (2001) Vive la différence: plant functional diversity matters to ecosystem processes. *Trends in Ecology & Evolution*, **16**, 646–655.
- Dykhuizen, D. E. & Hartl, D. L. (1983) Selection in chemostats.. *Microbiological reviews*, **47**, 150–168.
- Fridley, J. D. , Vandermast, D. B. , Kuppinger, D. M. , Manthey, M. , & Peet, R. K. (2007) Co-occurrence based assessment of habitat generalists and specialists: a new approach for the measurement of niche width. *Journal of Ecology*, **95**, 707–722.
- Gentleman, R. & Biocore (2009) geneplotter: Graphics related functions for Bioconductor. R package version 1.20.0.
- Gilchrist, G. W. (1995) Specialists and Generalists in Changing Environments. I. Fitness Landscapes of Thermal Sensitivity. *The American Naturalist*, **146**, 252–270.
- Hardesty, D. (1975) The niche concept: Suggestions for its use in human ecology. *Human Ecology*, **3**, 71–85.
- Levins, R. (1968) Evolution in Changing Environments: Some Theoretical Explorations. (MPB-2) *Monographs in Population Biology*.

Macarthur, R. H. & Wilson, E. O. (1967) The Theory of Island Biogeography.

NatureServe. 2009. NatureServe Explorer: An online encyclopedia of life [web application]. Version 7.1. NatureServe, Arlington, Virginia. Available <http://www.natureserve.org/explorer>. (Accessed: June 6, 2009).

Nekola, Jeffrey, C., Brown, & James, H. (2007) The wealth of species: ecological communities, complex systems and the legacy of Frank Preston. *Ecology Letters*, **10**, 188–196.

Oksanen, J., Kindt, R., Legendre, P., O'Hara, B., Simpson, G. L., Solymos, P., Stevens, M. H. H., & Wagner, H. (2008) vegan: Community Ecology Package. R package version 1.15-1.

Palmer, J. et al., 2007. Electronic Edition of Domesday Book: Translation, Databases and Scholarly Commentary, 1086. *Colchester, Essex: UK Data Archive*.

Peet, R.K. and Christensen N.L. (1988) Changes in species diversity during secondary forest succession on the North Carolina USA Piedmont. In: During HJ, Werger MJA and Willems HJ (eds) Diversity and Pattern in Plant Communities; International Symposium on Vegetational Structure, Woudschoten, Utrecht, Netherlands, 1987: 233–246. The Hague, Netherlands: SPB Academic Publishing BV.

Peet, R. K., Wentworth, T. R., Schafale, M. P., & Weakley, A. S. (2004) Carolina Vegetation Survey database. Version 3.0. North Carolina Botanical Garden. Chapel Hill, NC 27599.

R Development Core Team (2009) R: A Language and Environment for Statistical Computing. Version 2.8.1.

Ricklefs, R. E. (1987) Community Diversity: Relative Roles of Local and Regional Processes. *Science*, **235**, 167–171.

Sarkar, D. (2009) lattice: Lattice Graphics. R package version 0.17-20.

Tilman, D. (1982) Resource Competition and Community Structure.

Tilman, D. (1994) Competition and Biodiversity in Spatially Structured Habitats. *Ecology*, **75**, 2–16.

Wickham & Hadley (2007) Reshaping data with the reshape package. *Journal of Statistical Software*, **21**.

Postscript

Species generalism is multi-use tool on the ecologist's belt. It is a universal concept that can be applied to all species, and is easily related to more specific life history traits and strategies. By extension, it is related to a wide variety of ecological processes and concepts.

I have shown how it can be estimated with large-scale abundance data. The advantage of this is ease of use, by eliminating the need to identify and measure a myriad of relevant variables within a species' range, or alternately show its response experimentally. Manipulating abundance data in key ways helps to ensure estimates in a variety of systems.

The cost of generalism's universality is a coarseness. But its utility in summarizing species niches is extended greatly by considering generalism across spatial scales. As different events inherently have scale, the "multi-grain niche" allows inferences about species' sensitivities to those events.

Generalism can also be used to focus on such events themselves, rather than species; that is, on ecological processes. I have shown how generalism of a community's species can be taken as a whole, and how different patterns could result from the importance of broad types of process, like competition and evolutionary trade-offs. Future work might combine the thinking of chapters 2 and 3 to consider species'

multi-grain niches in the context of a community or landscape so as to infer the strength of different types of process.

Although the real world is made up of specifics, which in terms of information ultimately cannot be compressed, it is science's mission to simplify as the need for human comprehension of the world demands it. Species, localities, and community types must all be studied individually. But they must also be summarized, and for this, generalism shows great promise.

Appendices

APPENDIX A: CHAPTER 1 ANOVA RESULTS

		Df	Sum Sq	Mean Sq	F value	
	Pr(>F)					
A	1	21.049	21.049	4181.5487	< 2.2e-16	***
C	2	12.969	6.485	1288.2298	< 2.2e-16	***
D	2	19.900	9.950	1976.6325	< 2.2e-16	***
E	3	2.713	0.904	179.6842	< 2.2e-16	***
subsamp	1	3.681	3.681	731.3382	< 2.2e-16	***
posMethod	2	8.940	4.470	888.0003	< 2.2e-16	***
plotBias	1	6.077	6.077	1207.3113	< 2.2e-16	***
widthMethod	1	71.886	71.886	14280.9248	< 2.2e-16	***
meanWidth	1	107.587	107.587	21373.2958	< 2.2e-16	***
A:C	2	14.904	7.452	1480.4031	< 2.2e-16	***
A:D	2	2.234	1.117	221.9174	< 2.2e-16	***
C:D	4	0.375	0.094	18.6293	3.045e-15	***
A:E	3	2.342	0.781	155.0768	< 2.2e-16	***
C:E	6	0.111	0.018	3.6591	0.0012487	**
D:E	6	3.519	0.586	116.5006	< 2.2e-16	***
A:subsamp	1	0.756	0.756	150.2386	< 2.2e-16	***
C:subsamp	2	0.004	0.002	0.3939	0.6744163	
D:subsamp	2	3.222	1.611	320.0256	< 2.2e-16	***
E:subsamp	3	0.012	0.004	0.7758	0.5073300	
A:posMethod	2	0.353	0.177	35.0750	6.981e-16	***
C:posMethod	4	0.015	0.004	0.7261	0.5739748	
D:posMethod	4	0.061	0.015	3.0516	0.0159429	*
E:posMethod	6	1.385	0.231	45.8571	< 2.2e-16	***
subsamp:posMethod	2	0.304	0.152	30.1982	8.751e-14	***
A:plotBias	1	0.106	0.106	21.0757	4.493e-06	***
C:plotBias	2	0.007	0.004	0.7160	0.4887605	
D:plotBias	2	1.000	0.500	99.3686	< 2.2e-16	***
E:plotBias	3	0.029	0.010	1.9365	0.1213674	
subsamp:plotBias	1	0.701	0.701	139.2467	< 2.2e-16	***
posMethod:plotBias	2	0.051	0.025	5.0599	0.0063700	**
A:widthMethod	1	0.459	0.459	91.1148	< 2.2e-16	***
C:widthMethod	2	0.053	0.027	5.3079	0.0049727	**
D:widthMethod	2	2.150	1.075	213.5618	< 2.2e-16	***
E:widthMethod	3	0.036	0.012	2.3969	0.0661574	.
subsamp:widthMethod	1	0.138	0.138	27.4159	1.689e-07	***
posMethod:widthMethod	2	0.397	0.198	39.4054	< 2.2e-16	***
plotBias:widthMethod	1	0.135	0.135	26.8525	2.258e-07	***
A:meanWidth	1	0.030	0.030	5.8651	0.0154699	*
C:meanWidth	2	0.245	0.122	24.2995	3.046e-11	***
D:meanWidth	2	0.157	0.078	15.5918	1.753e-07	***
E:meanWidth	3	1.587	0.529	105.0901	< 2.2e-16	***
subsamp:meanWidth	1	0.040	0.040	7.9253	0.0048886	**
posMethod:meanWidth	2	2.942	1.471	292.2431	< 2.2e-16	***
plotBias:meanWidth	1	0.660	0.660	131.0722	< 2.2e-16	***
widthMethod:meanWidth	1	21.636	21.636	4298.2506	< 2.2e-16	***
A:C:D	4	0.101	0.025	4.9976	0.0005075	***
A:C:E	6	0.359	0.060	11.8859	2.585e-13	***
A:D:E	6	3.491	0.582	115.5849	< 2.2e-16	***
C:D:E	12	0.088	0.007	1.4595	0.1315665	
A:C:subsamp	2	0.146	0.073	14.5045	5.175e-07	***
A:D:subsamp	2	0.365	0.182	36.2081	2.274e-16	***
C:D:subsamp	4	0.016	0.004	0.7944	0.5286027	
A:E:subsamp	3	0.280	0.093	18.5735	5.368e-12	***
C:E:subsamp	6	0.017	0.003	0.5691	0.7552382	
D:E:subsamp	6	0.537	0.090	17.7908	< 2.2e-16	***
A:C:posMethod	4	0.091	0.023	4.5129	0.0012173	**
A:D:posMethod	4	0.235	0.059	11.6852	1.856e-09	***
C:D:posMethod	8	0.011	0.001	0.2739	0.9745736	
A:E:posMethod	6	0.120	0.020	3.9663	0.0005771	***
C:E:posMethod	12	0.005	4.435e-04	0.0881	0.9999806	
D:E:posMethod	12	0.259	0.022	4.2808	8.528e-07	***
A:subsamp:posMethod	2	0.008	0.004	0.7756	0.4604508	
C:subsamp:posMethod	4	0.002	3.980e-04	0.0791	0.9887373	
D:subsamp:posMethod	4	0.085	0.021	4.2441	0.0019696	**

E:subsamp:posMethod	6	0.037	0.006	1.2243	0.2901906	
A:C:plotBias	2	0.102	0.051	10.1181	4.095e-05	***
A:D:plotBias	2	0.013	0.007	1.3382	0.2623877	
C:D:plotBias	4	0.024	0.006	1.2006	0.3082903	
A:E:plotBias	3	0.033	0.011	2.2174	0.0839448	.
C:E:plotBias	6	0.001	2.313e-04	0.0459	0.9996059	
D:E:plotBias	6	0.075	0.012	2.4741	0.0215965	*
A:subsamp:plotBias	1	0.005	0.005	0.9805	0.3220981	
C:subsamp:plotBias	2	0.001	4.213e-04	0.0837	0.9197158	
D:subsamp:plotBias	2	0.018	0.009	1.8365	0.1594483	
E:subsamp:plotBias	3	0.011	0.004	0.6979	0.5532392	
A:posMethod:plotBias	2	0.015	0.008	1.4946	0.2244113	
C:posMethod:plotBias	4	0.003	0.001	0.1558	0.9604467	
D:posMethod:plotBias	4	0.037	0.009	1.8324	0.1195927	
E:posMethod:plotBias	6	0.019	0.003	0.6166	0.7172013	
subsamp:posMethod:plotBias	2	0.131	0.065	13.0021	2.311e-06	***
A:C:widthMethod	2	0.274	0.137	27.2105	1.694e-12	***
A:D:widthMethod	2	0.117	0.059	11.6524	8.870e-06	***
C:D:widthMethod	4	0.257	0.064	12.7733	2.328e-10	***
A:E:widthMethod	3	0.054	0.018	3.5781	0.0132987	*
C:E:widthMethod	6	0.021	0.003	0.6851	0.6616891	
D:E:widthMethod	6	0.038	0.006	1.2740	0.2655424	
A:subsamp:widthMethod	1	0.001	0.001	0.2508	0.6164961	
C:subsamp:widthMethod	2	0.037	0.018	3.6455	0.0261596	*
D:subsamp:widthMethod	2	0.232	0.116	23.0540	1.050e-10	***
E:subsamp:widthMethod	3	0.018	0.006	1.2032	0.3069139	***
A:posMethod:widthMethod	2	0.008	0.004	0.8098	0.4449778	
C:posMethod:widthMethod	4	0.037	0.009	1.8185	0.1222227	
D:posMethod:widthMethod	4	0.068	0.017	3.3622	0.0093234	**
E:posMethod:widthMethod	6	0.028	0.005	0.9370	0.4669151	
subsamp:posMethod:widthMethod	2	0.049	0.024	4.8520	0.0078390	**
A:plotBias:widthMethod	1	0.018	0.018	3.5489	0.0596279	.
C:plotBias:widthMethod	2	0.001	0.001	0.1064	0.8990674	
D:plotBias:widthMethod	2	0.005	0.002	0.4575	0.6328971	
E:plotBias:widthMethod	3	0.002	0.001	0.1638	0.9207939	
subsamp:plotBias:widthMethod	1	0.024	0.024	4.7909	0.0286440	*
posMethod:plotBias:widthMethod	2	0.026	0.013	2.5814	0.0757411	.
A:C:meanWidth	2	1.369	0.684	135.9623	< 2.2e-16	***
A:D:meanWidth	2	0.917	0.458	91.0810	< 2.2e-16	***
C:D:meanWidth	4	0.752	0.188	37.3552	< 2.2e-16	***
A:E:meanWidth	3	0.332	0.111	22.0128	3.502e-14	***
C:E:meanWidth	6	0.068	0.011	2.2534	0.0356028	*
D:E:meanWidth	6	0.717	0.120	23.7516	< 2.2e-16	***
A:subsamp:meanWidth	1	0.085	0.085	16.8436	4.106e-05	***
C:subsamp:meanWidth	2	0.045	0.023	4.4702	0.0114784	*
D:subsamp:meanWidth	2	0.341	0.171	33.8998	2.235e-15	***
E:subsamp:meanWidth	3	0.050	0.017	3.3197	0.0189738	*
A:posMethod:meanWidth	2	0.044	0.022	4.4151	0.0121275	*
C:posMethod:meanWidth	4	0.005	0.001	0.2241	0.9250699	
D:posMethod:meanWidth	4	0.082	0.020	4.0709	0.0026813	**
E:posMethod:meanWidth	6	0.441	0.073	14.5982	< 2.2e-16	***
subsamp:posMethod:meanWidth	2	0.106	0.053	10.5775	2.590e-05	***
A:plotBias:meanWidth	1	0.006	0.006	1.1007	0.2941387	
C:plotBias:meanWidth	2	0.001	3.332e-04	0.0662	0.9359447	
D:plotBias:meanWidth	2	0.234	0.117	23.2807	8.380e-11	***
E:plotBias:meanWidth	3	0.011	0.004	0.7081	0.5470553	
subsamp:plotBias:meanWidth	1	0.438	0.438	86.9476	< 2.2e-16	***
posMethod:plotBias:meanWidth	2	0.327	0.163	32.4325	9.560e-15	***
A:widthMethod:meanWidth	1	0.013	0.013	2.5145	0.1128467	
C:widthMethod:meanWidth	2	0.030	0.015	2.9354	0.0531752	.
D:widthMethod:meanWidth	2	0.340	0.170	33.7655	2.553e-15	***
E:widthMethod:meanWidth	3	0.046	0.015	3.0661	0.0268352	*
subsamp:widthMethod:meanWidth	1	0.095	0.095	18.9537	1.359e-05	***
posMethod:widthMethod:meanWidth	2	0.435	0.218	43.2474	< 2.2e-16	***
plotBias:widthMethod:meanWidth	1	0.033	0.033	6.5296	0.0106308	*
A:C:D:E	12	0.042	0.004	0.6977	0.7553394	
A:C:D:subsamp	4	0.031	0.008	1.5525	0.1841474	
A:C:E:subsamp	6	0.008	0.001	0.2703	0.9509622	
A:D:E:subsamp	6	0.357	0.059	11.8163	3.143e-13	***
C:D:E:subsamp	12	0.028	0.002	0.4580	0.9392384	
A:C:D:posMethod	8	0.007	0.001	0.1744	0.9943099	
A:C:E:posMethod	12	0.013	0.001	0.2097	0.9980976	
A:D:E:posMethod	12	0.154	0.013	2.5552	0.0022535	**
C:D:E:posMethod	24	0.007	2.930e-04	0.0582	1.0000000	
A:C:subsamp:posMethod	4	0.002	0.001	0.1154	0.9771080	
A:D:subsamp:posMethod	4	0.002	0.001	0.1176	0.9763293	
C:D:subsamp:posMethod	8	0.007	0.001	0.1756	0.9941758	
A:E:subsamp:posMethod	6	0.005	0.001	0.1633	0.9863626	
C:E:subsamp:posMethod	12	0.002	1.777e-04	0.0353	0.9999999	
D:E:subsamp:posMethod	12	0.043	0.004	0.7156	0.7376051	
A:C:D:plotBias	4	0.013	0.003	0.6580	0.6212107	

A:C:E:plotBias	6	0.004	0.001	0.1471	0.9896679
A:D:E:plotBias	6	0.030	0.005	0.9972	0.4251583
C:D:E:plotBias	12	0.001	9.557e-05	0.0190	1.0000000
A:C:subsamp:plotBias	2	0.015	0.008	1.5352	0.2154795
A:D:subsamp:plotBias	2	0.005	0.003	0.5425	0.5813334
C:D:subsamp:plotBias	4	0.002	0.001	0.1156	0.9770437
A:E:subsamp:plotBias	3	0.002	0.001	0.1307	0.9418830
C:E:subsamp:plotBias	6	0.001	9.299e-05	0.0185	0.9999728
D:E:subsamp:plotBias	6	0.006	0.001	0.1978	0.9775176
A:C:posMethod:plotBias	4	0.005	0.001	0.2378	0.9171195
A:D:posMethod:plotBias	4	0.063	0.016	3.1448	0.0135811 *
C:D:posMethod:plotBias	8	0.003	3.587e-04	0.0713	0.9997805
A:E:posMethod:plotBias	6	0.008	0.001	0.2701	0.9510532
C:E:posMethod:plotBias	12	0.001	9.452e-05	0.0188	1.0000000
D:E:posMethod:plotBias	12	0.012	0.001	0.2007	0.9984702
A:subsamp:posMethod:plotBias	2	0.005	0.002	0.4671	0.6268676
C:subsamp:posMethod:plotBias	4	0.002	4.979e-04	0.0989	0.9828271
D:subsamp:posMethod:plotBias	4	0.008	0.002	0.4079	0.8030962
E:subsamp:posMethod:plotBias	6	0.009	0.002	0.2999	0.9371551
A:C:D:widthMethod	4	0.061	0.015	3.0126	0.0170428 *
A:C:E:widthMethod	6	0.007	0.001	0.2168	0.9715853
A:D:E:widthMethod	6	0.052	0.009	1.7235	0.1111835
C:D:E:widthMethod	12	0.007	0.001	0.1116	0.9999289
A:C:subsamp:widthMethod	2	0.007	0.003	0.6476	0.5233499
A:D:subsamp:widthMethod	2	1.032e-04	5.158e-05	0.0102	0.9898054
C:D:subsamp:widthMethod	4	0.021	0.005	1.0639	0.3726278
A:E:subsamp:widthMethod	3	0.001	2.954e-04	0.0587	0.9813569
C:E:subsamp:widthMethod	6	0.002	2.583e-04	0.0513	0.9994575
D:E:subsamp:widthMethod	6	0.020	0.003	0.6553	0.6859262
A:C:posMethod:widthMethod	4	0.001	1.887e-04	0.0375	0.9973240
A:D:posMethod:widthMethod	4	0.027	0.007	1.3249	0.2580404
C:D:posMethod:widthMethod	8	0.006	0.001	0.1570	0.9960494
A:E:posMethod:widthMethod	6	0.009	0.002	0.3072	0.9335331
C:E:posMethod:widthMethod	12	0.003	2.467e-04	0.0490	0.9999993
D:E:posMethod:widthMethod	12	0.019	0.002	0.3082	0.9882571
A:subsamp:posMethod:widthMethod	2	0.015	0.007	1.4762	0.2285784
C:subsamp:posMethod:widthMethod	4	0.004	0.001	0.2045	0.9359825
D:subsamp:posMethod:widthMethod	4	0.010	0.002	0.4953	0.7392179
E:subsamp:posMethod:widthMethod	6	0.006	0.001	0.1866	0.9806598
A:C:plotBias:widthMethod	2	0.026	0.013	2.6128	0.0733989 .
A:D:plotBias:widthMethod	2	0.007	0.004	0.7084	0.4924537
C:D:plotBias:widthMethod	4	0.001	2.796e-04	0.0555	0.9942673
A:E:plotBias:widthMethod	3	0.002	0.001	0.1648	0.9201277
C:E:plotBias:widthMethod	6	8.293e-05	1.382e-05	0.0027	0.9999999
D:E:plotBias:widthMethod	6	0.004	0.001	0.1421	0.9905780
A:subsamp:plotBias:widthMethod	1	0.005	0.005	1.0894	0.2966406
C:subsamp:plotBias:widthMethod	2	0.001	0.001	0.1151	0.8912475
D:subsamp:plotBias:widthMethod	2	0.014	0.007	1.3726	0.2535082
E:subsamp:plotBias:widthMethod	3	0.006	0.002	0.3842	0.7643633
A:posMethod:plotBias:widthMethod	2	0.008	0.004	0.7491	0.4728071
C:posMethod:plotBias:widthMethod	4	0.001	1.306e-04	0.0259	0.9986995
D:posMethod:plotBias:widthMethod	4	0.048	0.012	2.3658	0.0506157 .
E:posMethod:plotBias:widthMethod	6	0.006	0.001	0.2147	0.9722984
subsamp:posMethod:plotBias:widthMethod	2	0.080	0.040	7.9606	0.0003522 ***
A:C:D:meanWidth	4	0.221	0.055	10.9595	7.385e-09 ***
A:C:E:meanWidth	6	0.020	0.003	0.6740	0.6707517
A:D:E:meanWidth	6	0.509	0.085	16.8670	< 2.2e-16 ***
C:D:E:meanWidth	12	0.142	0.012	2.3580	0.0050701 **
A:C:subsamp:meanWidth	2	8.621e-05	4.311e-05	0.0086	0.9914729
A:D:subsamp:meanWidth	2	0.122	0.061	12.1590	5.354e-06 ***
C:D:subsamp:meanWidth	4	0.022	0.006	1.1130	0.3483626
A:E:subsamp:meanWidth	3	0.070	0.023	4.6478	0.0030015 **
C:E:subsamp:meanWidth	6	0.013	0.002	0.4185	0.8671712
D:E:subsamp:meanWidth	6	0.203	0.034	6.7362	3.957e-07 ***
A:C:posMethod:meanWidth	4	0.013	0.003	0.6698	0.6129146
A:D:posMethod:meanWidth	4	0.065	0.016	3.2106	0.0121251 *
C:D:posMethod:meanWidth	8	0.010	0.001	0.2530	0.9802545
A:E:posMethod:meanWidth	6	0.058	0.010	1.9094	0.0754260 .
C:E:posMethod:meanWidth	12	0.003	2.776e-04	0.0552	0.9999986
D:E:posMethod:meanWidth	12	0.079	0.007	1.3129	0.2030315
A:subsamp:posMethod:meanWidth	2	0.005	0.003	0.5050	0.6035548
C:subsamp:posMethod:meanWidth	4	0.003	0.001	0.1251	0.9734612
D:subsamp:posMethod:meanWidth	4	0.058	0.015	2.9012	0.0206156 *
E:subsamp:posMethod:meanWidth	6	0.016	0.003	0.5385	0.7793095
A:C:plotBias:meanWidth	2	0.002	0.001	0.1897	0.8272498
A:D:plotBias:meanWidth	2	1.921e-04	9.605e-05	0.0191	0.9811005
C:D:plotBias:meanWidth	4	0.009	0.002	0.4717	0.7565613
A:E:plotBias:meanWidth	3	0.014	0.005	0.9419	0.4193244
C:E:plotBias:meanWidth	6	0.002	2.637e-04	0.0524	0.9994242
D:E:plotBias:meanWidth	6	0.041	0.007	1.3541	0.2292929
A:subsamp:plotBias:meanWidth	1	0.001	0.001	0.1689	0.6811366

C:subsamp:plotBias:meanWidth	2	7.588e-05	3.794e-05	0.0075	0.9924911	
D:subsamp:plotBias:meanWidth	2	0.014	0.007	1.3953	0.2478311	
E:subsamp:plotBias:meanWidth	3	0.009	0.003	0.6122	0.6070338	
A:posMethod:plotBias:meanWidth	2	0.007	0.003	0.6657	0.5139442	
C:posMethod:plotBias:meanWidth	4	8.776e-05	2.194e-05	0.0044	0.9999622	
D:posMethod:plotBias:meanWidth	4	0.022	0.006	1.1173	0.3462936	
E:posMethod:plotBias:meanWidth	6	0.010	0.002	0.3374	0.9173939	
subsamp:posMethod:plotBias:meanWidth	2	0.056	0.028	5.5676	0.0038368	**
A:C:widthMethod:meanWidth	2	0.023	0.011	2.2580	0.1046333	
A:D:widthMethod:meanWidth	2	0.004	0.002	0.4302	0.6504060	
C:D:widthMethod:meanWidth	4	0.021	0.005	1.0593	0.3749497	
A:E:widthMethod:meanWidth	3	0.011	0.004	0.6964	0.5541257	
C:E:widthMethod:meanWidth	6	0.001	1.187e-04	0.0236	0.9999440	
D:E:widthMethod:meanWidth	6	0.025	0.004	0.8376	0.5405598	
A:subsamp:widthMethod:meanWidth	1	0.026	0.026	5.0930	0.0240541	*
C:subsamp:widthMethod:meanWidth	2	0.008	0.004	0.7528	0.4710978	
D:subsamp:widthMethod:meanWidth	2	0.044	0.022	4.3731	0.0126470	*
E:subsamp:widthMethod:meanWidth	3	0.005	0.002	0.2985	0.8264869	
A:posMethod:widthMethod:meanWidth	2	0.012	0.006	1.1642	0.3122325	
C:posMethod:widthMethod:meanWidth	4	0.012	0.003	0.5983	0.6638716	
D:posMethod:widthMethod:meanWidth	4	0.015	0.004	0.7602	0.5510883	
E:posMethod:widthMethod:meanWidth	6	0.018	0.003	0.5909	0.7379393	
subsamp:posMethod:widthMethod:meanWidth	2	0.004	0.002	0.4421	0.6427354	
A:plotBias:widthMethod:meanWidth	1	4.511e-04	4.511e-04	0.0896	0.7646737	
C:plotBias:widthMethod:meanWidth	2	0.001	0.001	0.1134	0.8927852	
D:plotBias:widthMethod:meanWidth	2	0.001	0.001	0.1348	0.8738773	
E:plotBias:widthMethod:meanWidth	3	0.002	0.001	0.1497	0.9299383	
subsamp:plotBias:widthMethod:meanWidth	1	0.027	0.027	5.4176	0.0199636	*
posMethod:plotBias:widthMethod:meanWidth	2	0.052	0.026	5.1357	0.0059054	**
A:C:D:E:subsamp	12	0.012	0.001	0.2065	0.9982375	
A:C:D:E:posMethod	24	0.004	1.556e-04	0.0309	1.0000000	
A:C:D:subsamp:posMethod	8	0.001	1.415e-04	0.0281	0.9999939	
A:C:E:subsamp:posMethod	12	0.001	1.142e-04	0.0227	1.0000000	
A:D:E:subsamp:posMethod	12	0.006	4.909e-04	0.0975	0.9999660	
C:D:E:subsamp:posMethod	24	0.001	2.301e-05	0.0046	1.0000000	
A:C:D:E:plotBias	12	0.001	6.355e-05	0.0126	1.0000000	
A:C:D:subsamp:plotBias	4	0.001	2.636e-04	0.0524	0.9948820	
A:C:E:subsamp:plotBias	6	0.001	1.536e-04	0.0305	0.9998806	
A:D:E:subsamp:plotBias	6	4.519e-04	7.532e-05	0.0150	0.9999854	
C:D:E:subsamp:plotBias	12	2.348e-04	1.956e-05	0.0039	1.0000000	
A:C:D:posMethod:plotBias	8	0.003	3.276e-04	0.0651	0.9998442	
A:C:E:posMethod:plotBias	12	0.001	1.156e-04	0.0230	1.0000000	
A:D:E:posMethod:plotBias	12	0.016	0.001	0.2576	0.9948530	
C:D:E:posMethod:plotBias	24	0.001	5.561e-05	0.0110	1.0000000	
A:C:subsamp:posMethod:plotBias	4	0.010	0.003	0.5082	0.7297417	
A:D:subsamp:posMethod:plotBias	4	0.003	0.001	0.1503	0.9629332	
C:D:subsamp:posMethod:plotBias	8	2.799e-04	3.498e-05	0.0069	1.0000000	
A:E:subsamp:posMethod:plotBias	6	0.001	1.233e-04	0.0245	0.9999373	
C:E:subsamp:posMethod:plotBias	12	1.454e-04	1.212e-05	0.0024	1.0000000	
D:E:subsamp:posMethod:plotBias	12	0.003	2.682e-04	0.0533	0.9999989	
A:C:D:E:widthMethod	12	0.011	0.001	0.1779	0.9991677	
A:C:D:subsamp:widthMethod	4	0.007	0.002	0.3339	0.8553134	
A:C:E:subsamp:widthMethod	6	0.002	3.391e-04	0.0674	0.9988157	
A:D:E:subsamp:widthMethod	6	0.004	0.001	0.1324	0.9922213	
C:D:E:subsamp:widthMethod	12	0.003	2.764e-04	0.0549	0.9999987	
A:C:D:posMethod:widthMethod	8	0.004	0.001	0.1038	0.9991069	
A:C:E:posMethod:widthMethod	12	0.001	6.549e-05	0.0130	1.0000000	
A:D:E:posMethod:widthMethod	12	0.014	0.001	0.2350	0.9966814	
C:D:E:posMethod:widthMethod	24	0.001	4.034e-05	0.0080	1.0000000	
A:C:subsamp:posMethod:widthMethod	4	0.008	0.002	0.4185	0.7953996	
A:D:subsamp:posMethod:widthMethod	4	0.003	0.001	0.1274	0.9725461	
C:D:subsamp:posMethod:widthMethod	8	0.002	2.653e-04	0.0527	0.9999303	
A:E:subsamp:posMethod:widthMethod	6	0.002	2.636e-04	0.0524	0.9994248	
C:E:subsamp:posMethod:widthMethod	12	4.495e-04	3.746e-05	0.0074	1.0000000	
D:E:subsamp:posMethod:widthMethod	12	0.008	0.001	0.1362	0.9997930	
A:C:D:plotBias:widthMethod	4	0.002	4.703e-04	0.0934	0.9845650	
A:C:E:plotBias:widthMethod	6	4.574e-04	7.624e-05	0.0151	0.9999849	
A:D:E:plotBias:widthMethod	6	0.001	1.861e-04	0.0370	0.9997905	
C:D:E:plotBias:widthMethod	12	4.621e-04	3.851e-05	0.0077	1.0000000	
A:C:subsamp:plotBias:widthMethod	2	0.006	0.003	0.6190	0.5385206	
A:D:subsamp:plotBias:widthMethod	2	4.048e-04	2.024e-04	0.0402	0.9605902	
C:D:subsamp:plotBias:widthMethod	4	4.714e-04	1.178e-04	0.0234	0.9989372	
A:E:subsamp:plotBias:widthMethod	3	0.002	0.001	0.1494	0.9300842	
C:E:subsamp:plotBias:widthMethod	6	3.015e-04	5.025e-05	0.0100	0.9999956	
D:E:subsamp:plotBias:widthMethod	6	0.004	0.001	0.1452	0.9900290	
A:C:posMethod:plotBias:widthMethod	4	0.001	1.672e-04	0.0332	0.9978883	
A:D:posMethod:plotBias:widthMethod	4	0.012	0.003	0.5977	0.6642745	
C:D:posMethod:plotBias:widthMethod	8	0.002	2.066e-04	0.0410	0.9999734	
A:E:posMethod:plotBias:widthMethod	6	4.294e-04	7.157e-05	0.0142	0.9999875	
C:E:posMethod:plotBias:widthMethod	12	1.464e-04	1.220e-05	0.0024	1.0000000	
D:E:posMethod:plotBias:widthMethod	12	0.003	2.521e-04	0.0501	0.9999992	

A:subsamp:posMethod:plotBias:widthMethod	2	2.701e-04	1.350e-04	0.0268	0.9735278
C:subsamp:posMethod:plotBias:widthMethod	4	1.469e-04	3.673e-05	0.0073	0.9998945
D:subsamp:posMethod:plotBias:widthMethod	4	0.007	0.002	0.3682	0.8314565
E:subsamp:posMethod:plotBias:widthMethod	6	0.001	1.492e-04	0.0296	0.9998902
A:C:D:E:meanWidth	12	0.081	0.007	1.3458	0.1848091
A:C:D:subsamp:meanWidth	4	0.006	0.002	0.3163	0.8672515
A:C:E:subsamp:meanWidth	6	0.005	0.001	0.1693	0.9849985
A:D:E:subsamp:meanWidth	6	0.106	0.018	3.5018	0.0018459 **
C:D:E:subsamp:meanWidth	12	0.006	4.681e-04	0.0930	0.9999739
A:C:D:posMethod:meanWidth	8	0.004	0.001	0.1029	0.9991352
A:C:E:posMethod:meanWidth	12	0.004	3.020e-04	0.0600	0.9999978
A:D:E:posMethod:meanWidth	12	0.084	0.007	1.3886	0.1630227
C:D:E:posMethod:meanWidth	24	0.003	1.066e-04	0.0212	1.0000000
A:C:subsamp:posMethod:meanWidth	4	0.002	0.001	0.1166	0.9766683
A:D:subsamp:posMethod:meanWidth	4	0.001	1.600e-04	0.0318	0.9980631
C:D:subsamp:posMethod:meanWidth	8	0.003	4.268e-04	0.0848	0.9995783
A:E:subsamp:posMethod:meanWidth	6	0.004	0.001	0.1237	0.9943574
C:E:subsamp:posMethod:meanWidth	12	0.001	1.100e-04	0.0219	1.0000000
D:E:subsamp:posMethod:meanWidth	12	0.017	0.001	0.2814	0.9922377
A:C:D:plotBias:meanWidth	4	0.001	2.789e-04	0.0554	0.9942933
A:C:E:plotBias:meanWidth	6	0.001	1.648e-04	0.0327	0.9998531
A:D:E:plotBias:meanWidth	6	0.014	0.002	0.4629	0.8361613
C:D:E:plotBias:meanWidth	12	0.001	5.799e-05	0.0115	1.0000000
A:C:subsamp:plotBias:meanWidth	2	0.004	0.002	0.3958	0.6731775
A:D:subsamp:plotBias:meanWidth	2	0.004	0.002	0.4303	0.6503435
C:D:subsamp:plotBias:meanWidth	4	0.001	2.835e-04	0.0563	0.9941098
A:E:subsamp:plotBias:meanWidth	3	0.001	3.665e-04	0.0728	0.9745611
C:E:subsamp:plotBias:meanWidth	6	4.064e-04	6.773e-05	0.0135	0.9999894
D:E:subsamp:plotBias:meanWidth	6	0.004	0.001	0.1335	0.9920381
A:C:posMethod:plotBias:meanWidth	4	0.004	0.001	0.2192	0.9278490
A:D:posMethod:plotBias:meanWidth	4	0.035	0.009	1.7486	0.1363195
C:D:posMethod:plotBias:meanWidth	8	0.002	2.022e-04	0.0402	0.9999755
A:E:posMethod:plotBias:meanWidth	6	0.004	0.001	0.1168	0.9944675
C:E:posMethod:plotBias:meanWidth	12	4.567e-04	3.806e-05	0.0076	1.0000000
D:E:posMethod:plotBias:meanWidth	12	0.006	4.814e-04	0.0956	0.9999695
A:subsamp:posMethod:plotBias:meanWidth	2	0.003	0.001	0.2630	0.7687609
C:subsamp:posMethod:plotBias:meanWidth	4	0.001	2.748e-04	0.0546	0.9944561
D:subsamp:posMethod:plotBias:meanWidth	4	0.005	0.001	0.2462	0.9121041
E:subsamp:posMethod:plotBias:meanWidth	6	0.003	4.271e-04	0.0848	0.9977245
A:C:D:widthMethod:meanWidth	4	0.016	0.004	0.8081	0.5197411
A:C:E:widthMethod:meanWidth	6	0.005	0.001	0.1550	0.9881293
A:D:E:widthMethod:meanWidth	6	0.021	0.003	0.6951	0.6536426
C:D:E:widthMethod:meanWidth	12	0.002	1.838e-04	0.0365	0.9999999
A:C:subsamp:widthMethod:meanWidth	2	0.005	0.002	0.4821	0.6174937
A:D:subsamp:widthMethod:meanWidth	2	0.025	0.012	2.4355	0.0876308
C:D:subsamp:widthMethod:meanWidth	4	0.005	0.001	0.2604	0.9034089
A:E:subsamp:widthMethod:meanWidth	3	0.002	0.001	0.1349	0.9392405
C:E:subsamp:widthMethod:meanWidth	6	0.001	1.865e-04	0.0371	0.9997891
D:E:subsamp:widthMethod:meanWidth	6	0.004	0.001	0.1407	0.9908375
A:C:posMethod:widthMethod:meanWidth	4	0.001	2.129e-04	0.0423	0.9966168
A:D:posMethod:widthMethod:meanWidth	4	0.015	0.004	0.7553	0.5543359
C:D:posMethod:widthMethod:meanWidth	8	0.004	0.001	0.1100	0.9988975
A:E:posMethod:widthMethod:meanWidth	6	0.003	0.001	0.1097	0.9953495
C:E:posMethod:widthMethod:meanWidth	12	0.002	1.275e-04	0.0253	1.0000000
D:E:posMethod:widthMethod:meanWidth	12	0.010	0.001	0.1675	0.9993881
A:subsamp:posMethod:widthMethod:meanWidth	2	4.785e-04	2.393e-04	0.0475	0.9535817
C:subsamp:posMethod:widthMethod:meanWidth	4	0.002	4.353e-04	0.0865	0.9866591
D:subsamp:posMethod:widthMethod:meanWidth	4	0.017	0.004	0.8211	0.5114491
E:subsamp:posMethod:widthMethod:meanWidth	6	0.002	2.830e-04	0.0562	0.9992943
A:C:plotBias:widthMethod:meanWidth	2	0.003	0.001	0.2878	0.7499265
A:D:plotBias:widthMethod:meanWidth	2	0.004	0.002	0.4377	0.6455481
C:D:plotBias:widthMethod:meanWidth	4	0.002	4.280e-04	0.0850	0.9870775
A:E:plotBias:widthMethod:meanWidth	3	0.001	2.300e-04	0.0457	0.9870465
C:E:plotBias:widthMethod:meanWidth	6	5.151e-05	8.585e-06	0.0017	1.0000000
D:E:plotBias:widthMethod:meanWidth	6	0.002	3.670e-04	0.0729	0.9985170
A:subsamp:plotBias:widthMethod:meanWidth	1	0.003	0.003	0.6522	0.4193556
C:subsamp:plotBias:widthMethod:meanWidth	2	4.605e-04	2.303e-04	0.0457	0.9552859
D:subsamp:plotBias:widthMethod:meanWidth	2	0.008	0.004	0.8038	0.4476815
E:subsamp:plotBias:widthMethod:meanWidth	3	0.004	0.001	0.2838	0.8371514
A:posMethod:plotBias:widthMethod:meanWidth	2	0.001	0.001	0.1176	0.8890435
C:posMethod:plotBias:widthMethod:meanWidth	4	0.001	1.621e-04	0.0322	0.9980133
D:posMethod:plotBias:widthMethod:meanWidth	4	0.012	0.003	0.5744	0.6812259
E:posMethod:plotBias:widthMethod:meanWidth	6	0.001	2.272e-04	0.0451	0.9996257
subsamp:posMethod:plotBias:widthMethod:meanWidth	2	0.040	0.020	3.9440	0.0194143 *
A:C:D:E:subsamp:posMethod	24	0.001	3.086e-05	0.0061	1.0000000
A:C:D:E:subsamp:plotBias	12	9.762e-05	8.135e-06	0.0016	1.0000000
A:C:D:E:posMethod:plotBias	24	0.001	2.087e-05	0.0041	1.0000000
A:C:D:subsamp:posMethod:plotBias	8	4.295e-04	5.369e-05	0.0107	0.9999999
A:C:E:subsamp:posMethod:plotBias	12	0.001	7.476e-05	0.0149	1.0000000
A:D:E:subsamp:posMethod:plotBias	12	0.003	2.727e-04	0.0542	0.9999988
C:D:E:subsamp:posMethod:plotBias	24	1.694e-04	7.059e-06	0.0014	1.0000000

A:C:D:E:subsamp:widthMethod	12	0.002	2.046e-04	0.0407	0.99999998
A:C:D:E:posMethod:widthMethod	24	0.002	9.947e-05	0.0198	1.00000000
A:C:D:subsamp:posMethod:widthMethod	8	0.001	7.755e-05	0.0154	0.99999994
A:C:E:subsamp:posMethod:widthMethod	12	2.658e-04	2.215e-05	0.0044	1.00000000
A:D:E:subsamp:posMethod:widthMethod	12	0.001	1.115e-04	0.0222	1.00000000
C:D:E:subsamp:posMethod:widthMethod	24	4.005e-04	1.669e-05	0.0033	1.00000000
A:C:D:E:plotBias:widthMethod	12	2.533e-04	2.111e-05	0.0042	1.00000000
A:C:D:subsamp:plotBias:widthMethod	4	3.205e-04	8.012e-05	0.0159	0.9995038
A:C:E:subsamp:plotBias:widthMethod	6	0.001	1.089e-04	0.0216	0.9999566
A:D:E:subsamp:plotBias:widthMethod	6	0.001	1.764e-04	0.0351	0.9998207
C:D:E:subsamp:plotBias:widthMethod	12	9.898e-05	8.248e-06	0.0016	1.00000000
A:C:D:posMethod:plotBias:widthMethod	8	0.001	7.129e-05	0.0142	0.99999996
A:C:E:posMethod:plotBias:widthMethod	12	0.001	7.558e-05	0.0150	1.00000000
A:D:E:posMethod:plotBias:widthMethod	12	0.002	2.044e-04	0.0406	0.9999998
C:D:E:posMethod:plotBias:widthMethod	24	0.001	2.094e-05	0.0042	1.00000000
A:C:subsamp:posMethod:plotBias:widthMethod	4	0.001	3.192e-04	0.0634	0.9926030
A:D:subsamp:posMethod:plotBias:widthMethod	4	4.846e-04	1.212e-04	0.0241	0.9987777
C:D:subsamp:posMethod:plotBias:widthMethod	8	4.131e-04	5.164e-05	0.0103	0.9999999
A:E:subsamp:posMethod:plotBias:widthMethod	6	3.350e-04	5.583e-05	0.0111	0.9999940
C:E:subsamp:posMethod:plotBias:widthMethod	12	6.014e-05	5.012e-06	0.0010	1.00000000
D:E:subsamp:posMethod:plotBias:widthMethod	12	0.001	5.786e-05	0.0115	1.00000000
A:C:D:E:subsamp:meanWidth	12	0.001	1.112e-04	0.0221	1.00000000
A:C:D:E:posMethod:meanWidth	24	0.002	8.000e-05	0.0159	1.00000000
A:C:D:subsamp:posMethod:meanWidth	8	3.140e-04	3.925e-05	0.0078	1.00000000
A:C:E:subsamp:posMethod:meanWidth	12	0.001	4.979e-05	0.0099	1.00000000
A:D:E:subsamp:posMethod:meanWidth	12	0.005	3.816e-04	0.0758	0.9999916
C:D:E:subsamp:posMethod:meanWidth	24	4.258e-04	1.774e-05	0.0035	1.00000000
A:C:D:E:plotBias:meanWidth	12	3.282e-04	2.735e-05	0.0054	1.00000000
A:C:D:subsamp:plotBias:meanWidth	4	4.340e-04	1.085e-04	0.0216	0.9990967
A:C:E:subsamp:plotBias:meanWidth	6	4.721e-04	7.868e-05	0.0156	0.9999834
A:D:E:subsamp:plotBias:meanWidth	6	2.395e-04	3.992e-05	0.0079	0.9999978
C:D:E:subsamp:plotBias:meanWidth	12	8.629e-05	7.191e-06	0.0014	1.00000000
A:C:D:posMethod:plotBias:meanWidth	8	0.001	1.810e-04	0.0360	0.9999841
A:C:E:posMethod:plotBias:meanWidth	12	0.001	4.973e-05	0.0099	1.00000000
A:D:E:posMethod:plotBias:meanWidth	12	0.006	4.685e-04	0.0931	0.9999737
C:D:E:posMethod:plotBias:meanWidth	24	0.001	2.117e-05	0.0042	1.00000000
A:C:subsamp:posMethod:plotBias:meanWidth	4	0.004	0.001	0.1979	0.9395682
A:D:subsamp:posMethod:plotBias:meanWidth	4	3.077e-04	7.693e-05	0.0153	0.9995421
C:D:subsamp:posMethod:plotBias:meanWidth	8	7.007e-05	8.758e-06	0.0017	1.00000000
A:E:subsamp:posMethod:plotBias:meanWidth	6	4.906e-04	8.176e-05	0.0162	0.9999814
C:E:subsamp:posMethod:plotBias:meanWidth	12	4.695e-05	3.913e-06	0.0008	1.00000000
D:E:subsamp:posMethod:plotBias:meanWidth	12	0.002	1.844e-04	0.0366	0.9999999
A:C:D:E:widthMethod:meanWidth	12	0.001	1.135e-04	0.0225	1.00000000
A:C:D:subsamp:widthMethod:meanWidth	4	0.002	0.001	0.1229	0.9743110
A:C:E:subsamp:widthMethod:meanWidth	6	0.001	9.122e-05	0.0181	0.9999743
A:D:E:subsamp:widthMethod:meanWidth	6	0.002	2.609e-04	0.0518	0.9994415
C:D:E:subsamp:widthMethod:meanWidth	12	4.693e-04	3.911e-05	0.0078	1.00000000
A:C:D:posMethod:widthMethod:meanWidth	8	3.807e-04	4.759e-05	0.0095	0.9999999
A:C:E:posMethod:widthMethod:meanWidth	12	2.196e-04	1.830e-05	0.0036	1.00000000
A:D:E:posMethod:widthMethod:meanWidth	12	0.005	4.556e-04	0.0905	0.9999775
C:D:E:posMethod:widthMethod:meanWidth	24	0.001	2.128e-05	0.0042	1.00000000
A:C:subsamp:posMethod:widthMethod:meanWidth	4	1.116e-04	2.789e-05	0.0055	0.9999390
A:D:subsamp:posMethod:widthMethod:meanWidth	4	0.001	1.986e-04	0.0395	0.9970455
C:D:subsamp:posMethod:widthMethod:meanWidth	8	0.004	4.939e-04	0.0981	0.9992751
A:E:subsamp:posMethod:widthMethod:meanWidth	6	0.001	1.423e-04	0.0283	0.9999045
C:E:subsamp:posMethod:widthMethod:meanWidth	12	2.492e-04	2.077e-05	0.0041	1.00000000
D:E:subsamp:posMethod:widthMethod:meanWidth	12	0.003	2.508e-04	0.0498	0.9999992
A:C:D:plotBias:widthMethod:meanWidth	4	4.214e-04	1.053e-04	0.0209	0.9991480
A:C:E:plotBias:widthMethod:meanWidth	6	7.333e-05	1.222e-05	0.0024	0.9999999
A:D:E:plotBias:widthMethod:meanWidth	6	0.001	1.065e-04	0.0212	0.9999593
C:D:E:plotBias:widthMethod:meanWidth	12	2.571e-04	2.142e-05	0.0043	1.00000000
A:C:subsamp:plotBias:widthMethod:meanWidth	2	0.003	0.002	0.3179	0.7276799
A:D:subsamp:plotBias:widthMethod:meanWidth	2	4.549e-05	2.274e-05	0.0045	0.9954920
C:D:subsamp:plotBias:widthMethod:meanWidth	4	5.613e-05	1.403e-05	0.0028	0.9999845
A:E:subsamp:plotBias:widthMethod:meanWidth	3	0.001	4.696e-04	0.0933	0.9637604
C:E:subsamp:plotBias:widthMethod:meanWidth	6	2.421e-04	4.036e-05	0.0080	0.9999977
D:E:subsamp:plotBias:widthMethod:meanWidth	6	0.003	4.876e-04	0.0969	0.9967024
A:C:posMethod:plotBias:widthMethod:meanWidth	4	0.004	0.001	0.1830	0.9473329
A:D:posMethod:plotBias:widthMethod:meanWidth	4	0.004	0.001	0.1852	0.9462120
C:D:posMethod:plotBias:widthMethod:meanWidth	8	0.001	1.655e-04	0.0329	0.9999887
A:E:posMethod:plotBias:widthMethod:meanWidth	6	3.841e-04	6.402e-05	0.0127	0.9999910
C:E:posMethod:plotBias:widthMethod:meanWidth	12	5.438e-05	4.532e-06	0.0009	1.00000000
D:E:posMethod:plotBias:widthMethod:meanWidth	12	0.002	1.723e-04	0.0342	0.9999999
A:subsamp:posMethod:plotBias:widthMethod:meanWidth	2	1.370e-04	6.849e-05	0.0136	0.9864868
C:subsamp:posMethod:plotBias:widthMethod:meanWidth	4	1.817e-04	4.543e-05	0.0090	0.9998390
D:subsamp:posMethod:plotBias:widthMethod:meanWidth	4	0.007	0.002	0.3691	0.8308123
E:subsamp:posMethod:plotBias:widthMethod:meanWidth	6	0.001	1.299e-04	0.0258	0.9999269
A:C:D:E:subsamp:posMethod:plotBias	24	2.750e-04	1.146e-05	0.0023	1.00000000
A:C:D:E:subsamp:posMethod:plotBias	24	1.353e-04	5.638e-06	0.0011	1.00000000
A:C:D:E:subsamp:plotBias:widthMethod	12	6.521e-05	5.434e-06	0.0011	1.00000000
A:C:D:E:posMethod:plotBias:widthMethod	24	2.631e-04	1.096e-05	0.0022	1.00000000

A:C:D:subsamp:posMethod:plotBias:widthMethod	8	8.878e-05	1.110e-05	0.0022	1.0000000
A:C:E:subsamp:posMethod:plotBias:widthMethod	12	1.707e-04	1.422e-05	0.0028	1.0000000
A:D:E:subsamp:posMethod:plotBias:widthMethod	12	0.001	6.584e-05	0.0131	1.0000000
C:D:E:subsamp:posMethod:plotBias:widthMethod	24	7.097e-05	2.957e-06	0.0006	1.0000000
A:C:D:E:subsamp:posMethod:meanWidth	24	3.646e-04	1.519e-05	0.0030	1.0000000
A:C:D:E:subsamp:plotBias:meanWidth	12	5.786e-05	4.821e-06	0.0010	1.0000000
A:C:D:E:posMethod:plotBias:meanWidth	24	1.836e-04	7.649e-06	0.0015	1.0000000
A:C:D:subsamp:posMethod:plotBias:meanWidth	8	1.850e-04	2.312e-05	0.0046	1.0000000
A:C:E:subsamp:posMethod:plotBias:meanWidth	12	2.379e-04	1.983e-05	0.0039	1.0000000
A:D:E:subsamp:posMethod:plotBias:meanWidth	12	0.002	1.483e-04	0.0295	1.0000000
C:D:E:subsamp:posMethod:plotBias:meanWidth	24	4.996e-05	2.082e-06	0.0004	1.0000000
A:C:D:E:subsamp:widthMethod:meanWidth	12	2.601e-04	2.167e-05	0.0043	1.0000000
A:C:D:E:posMethod:widthMethod:meanWidth	24	0.001	2.944e-05	0.0058	1.0000000
A:C:D:subsamp:posMethod:widthMethod:meanWidth	8	2.414e-04	3.018e-05	0.0060	1.0000000
A:C:E:subsamp:posMethod:widthMethod:meanWidth	12	1.987e-04	1.656e-05	0.0033	1.0000000
A:D:E:subsamp:posMethod:widthMethod:meanWidth	12	0.001	5.942e-05	0.0118	1.0000000
C:D:E:subsamp:posMethod:widthMethod:meanWidth	24	1.707e-04	7.114e-06	0.0014	1.0000000
A:C:D:E:plotBias:widthMethod:meanWidth	12	1.120e-04	9.330e-06	0.0019	1.0000000
A:C:D:subsamp:plotBias:widthMethod:meanWidth	4	1.670e-05	4.175e-06	0.0008	0.9999986
A:C:E:subsamp:plotBias:widthMethod:meanWidth	6	3.804e-04	6.341e-05	0.0126	0.9999912
A:D:E:subsamp:plotBias:widthMethod:meanWidth	6	0.001	1.263e-04	0.0251	0.9999328
C:D:E:subsamp:plotBias:widthMethod:meanWidth	12	7.156e-05	5.964e-06	0.0012	1.0000000
A:C:D:posMethod:plotBias:widthMethod:meanWidth	8	3.842e-04	4.802e-05	0.0095	0.9999999
A:C:E:posMethod:plotBias:widthMethod:meanWidth	12	2.718e-04	2.265e-05	0.0045	1.0000000
A:D:E:posMethod:plotBias:widthMethod:meanWidth	12	0.001	9.499e-05	0.0189	1.0000000
C:D:E:posMethod:plotBias:widthMethod:meanWidth	24	1.223e-04	5.094e-06	0.0010	1.0000000
A:C:subsamp:posMethod:plotBias:widthMethod:meanWidth	4	3.232e-04	8.080e-05	0.0161	0.9994954
A:D:subsamp:posMethod:plotBias:widthMethod:meanWidth	4	3.179e-04	7.947e-05	0.0158	0.9995117
C:D:subsamp:posMethod:plotBias:widthMethod:meanWidth	8	1.189e-04	1.486e-05	0.0030	1.0000000
A:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	6	2.381e-04	3.969e-05	0.0079	0.9999978
C:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	12	7.386e-05	6.155e-06	0.0012	1.0000000
D:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	12	0.001	4.180e-05	0.0083	1.0000000
A:C:D:E:subsamp:posMethod:plotBias:widthMethod	24	1.043e-04	4.346e-06	0.0009	1.0000000
A:C:D:E:subsamp:posMethod:plotBias:meanWidth	24	7.781e-05	3.242e-06	0.0006	1.0000000
A:C:D:E:subsamp:posMethod:widthMethod:meanWidth	24	1.815e-04	7.563e-06	0.0015	1.0000000
A:C:D:E:subsamp:plotBias:widthMethod:meanWidth	12	4.297e-05	3.581e-06	0.0007	1.0000000
A:C:D:E:posMethod:plotBias:widthMethod:meanWidth	24	9.419e-05	3.924e-06	0.0008	1.0000000
A:C:D:subsamp:posMethod:plotBias:widthMethod:meanWidth	8	0.001	6.318e-05	0.0126	0.9999997
A:C:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	12	2.472e-05	2.060e-06	0.0004	1.0000000
A:D:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	12	4.225e-04	3.521e-05	0.0070	1.0000000
C:D:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	24	6.154e-05	2.564e-06	0.0005	1.0000000
A:C:D:E:subsamp:posMethod:plotBias:widthMethod:meanWidth	24	8.101e-05	3.375e-06	0.0007	1.0000000
Residuals	6912	34.793	0.005		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

APPENDIX B: CVS COMMUNITY-MODEL MATCHING

Model A

comm	scientific.name	nplots	rich
CEGL003570		9	77.4
CEGL003573		5	85.8
	Pinus palustris - Pinus taeda / Quercus geminata - Quercus hemisphaerica - Osmanthus americanus var. americanus /		
CEGL003577	Aristida stricta Woodland	5	30.2
CEGL003578		18	65.6
CEGL003583		5	43.4
	Pinus palustris / Quercus laevis / Aristida stricta / Cladonia spp. Woodland		
CEGL003584		11	21.6
CEGL003586		20	35.2
	Pinus palustris / Quercus laevis - Quercus geminata /		
CEGL003589	Vaccinium tenellum / Aristida stricta Woodland	7	23.9
CEGL003590		29	28.4
CEGL003593		12	79.7
CEGL003648		36	42.3
CEGL003660		7	103.0
	Pinus palustris - Pinus taeda - Pinus serotina / Quercus marilandica / (Quercus pumila) / Aristida stricta Woodland		
CEGL003664		7	93.4
	Pinus rigida - Quercus alba / Sporobolus heterolepis -		
CEGL003768	Andropogon gerardii Woodland	5	62.2
CEGL003881		3	29.7
CEGL003895	Alnus serrulata - Xanthorhiza simplicissima Shrubland	5	145.2
	Uniola paniculata - Schizachyrium littorale - Panicum amarum Herbaceous Vegetation		
CEGL004039		5	11.4
CEGL004073		3	49.7
CEGL004083		20	80.1
CEGL004084		17	87.4
CEGL004085		4	101.3
CEGL004418		3	69.3
CEGL004485		14	82.8
CEGL004487		4	101.8
CEGL004488		8	99.0
CEGL004489		4	60.5
CEGL004490		4	55.0
CEGL004492		3	51.7
CEGL004495		3	115.0
CEGL004499		4	93.3
CEGL004501		15	86.3
	Quercus laurifolia - Quercus michauxii - Liquidambar styraciflua / Carpinus caroliniana Forest		
CEGL004678		7	60.7
CEGL004691	Platanus occidentalis - Liriodendron tulipifera - Betula	7	100.1

	(alleghaniensis, lenta) / <i>Alnus serrulata</i> - <i>Leucothoe fontanesiana</i> Forest		
CEGL004733	<i>Taxodium distichum</i> - <i>Nyssa biflora</i> / <i>Fraxinus caroliniana</i> / <i>Lyonia lucida</i> Forest	4	33.3
CEGL004735	<i>Quercus lyrata</i> - <i>Quercus laurifolia</i> - <i>Taxodium distichum</i> / <i>Saururus cernuus</i> Forest	5	27.0
CEGL004737	<i>Quercus laurifolia</i> - <i>Quercus lyrata</i> / <i>Carpinus caroliniana</i> - <i>Persea palustris</i> / <i>Vaccinium elliotii</i> Forest	4	54.5
CEGL004788	<i>Quercus hemisphaerica</i> - <i>Magnolia grandiflora</i> - <i>Carya (glabra, pallida)</i> / <i>Vaccinium arboreum</i> / <i>Chasmanthium sessiliflorum</i> Forest	3	82.3
CEGL004985	<i>Pinus rigida</i> - (<i>Pinus pungens</i>) / <i>Rhododendron catawbiense</i> - <i>Kalmia latifolia</i> / <i>Galax urceolata</i> Woodland	3	29.7
CEGL005033		4	60.3
CEGL006130	<i>Fagus grandifolia</i> / <i>Carex pensylvanica</i> - <i>Carex brunnescens</i> Forest	5	48.4
CEGL006603		5	65.8
CEGL007022		3	32.7
CEGL007032	<i>Quercus virginiana</i> - (<i>Pinus elliotii</i> var. <i>elliotii</i> , <i>Sabal palmetto</i>) / <i>Persea borbonia</i> - <i>Callicarpa americana</i> Forest	9	31.3
CEGL007097	<i>Pinus pungens</i> - <i>Pinus rigida</i> - (<i>Quercus prinus</i>) / <i>Kalmia latifolia</i> - <i>Vaccinium pallidum</i> Woodland	19	35.8
CEGL007119	<i>Pinus virginiana</i> - <i>Pinus (rigida, echinata)</i> - (<i>Quercus prinus</i>) / <i>Vaccinium pallidum</i> Forest	7	36.4
CEGL007285	<i>Betula alleghaniensis</i> - <i>Fagus grandifolia</i> - <i>Aesculus flava</i> / <i>Viburnum lantanoides</i> / <i>Eurybia chlorolepis</i> - <i>Dryopteris intermedia</i> Forest	21	51.0
CEGL007291	<i>Liriodendron tulipifera</i> - <i>Tilia americana</i> var. <i>heterophylla</i> - (<i>Aesculus flava</i>) / <i>Actaea racemosa</i> Forest	13	48.0
CEGL007295	<i>Quercus alba</i> / <i>Kalmia latifolia</i> Forest	7	46.1
CEGL007298	<i>Quercus rubra</i> / <i>Carex pensylvanica</i> - <i>Ageratina altissima</i> var. <i>roanensis</i> Forest	17	46.2
CEGL007300	<i>Quercus rubra</i> / (<i>Vaccinium simulatum</i> , <i>Rhododendron calendulaceum</i>) / (<i>Dennstaedtia punctilobula</i> , <i>Thelypteris noveboracensis</i>) Forest	49	59.1
CEGL007316		5	102.8
CEGL007340		4	60.0
CEGL007356	<i>Quercus pagoda</i> - <i>Quercus phellos</i> - <i>Quercus lyrata</i> - <i>Quercus michauxii</i> / <i>Chasmanthium latifolium</i> Forest	6	70.2
CEGL007432	<i>Taxodium distichum</i> - <i>Nyssa aquatica</i> - <i>Nyssa biflora</i> / <i>Fraxinus caroliniana</i> / <i>Itea virginica</i> Forest	4	29.5
CEGL007540		9	49.8
CEGL007692	<i>Quercus alba</i> - <i>Quercus rubra</i> - <i>Quercus prinus</i> / <i>Collinsonia canadensis</i> - <i>Podophyllum peltatum</i> - <i>Amphicarpaea bracteata</i> Forest	10	69.6
CEGL007738		12	93.2
CEGL007844		8	42.4
CEGL007849	<i>Quercus pagoda</i> - <i>Quercus michauxii</i> - <i>Quercus alba</i> / <i>Arundinaria gigantea</i> ssp. <i>tecta</i> - <i>Sabal minor</i> / <i>Chasmanthium laxum</i> Forest	3	70.3
CEGL007850		4	45.5

CEGL007861	Betula alleghaniensis - (Tsuga canadensis) / Rhododendron maximum / Leucothoe fontanesiana Forest	9	30.1
CEGL008522		5	65.4

Model B

comm	scientific.name	nplots	rich
CEGL003591		10	35.5
CEGL003599		8	41.4
CEGL004496		27	91.5
CEGL004502		6	83.5
CEGL004740	Celtis laevigata - Fraxinus pennsylvanica - Acer negundo - (Juglans nigra) / Asimina triloba / Carex grayi Forest	9	46.1
CEGL004973	Aesculus flava - Betula alleghaniensis - Acer saccharum / Acer spicatum / Caulophyllum thalictroides - Laportea canadensis Forest	29	41.9
CEGL006192	Quercus rubra - Acer rubrum / Calycanthus floridus - Pyrus pubera / Thelypteris noveboracensis Forest	32	62.5
CEGL006286	Quercus prinus - Quercus rubra / Rhododendron maximum / Galax urceolata Forest	12	33.3
CEGL007027		11	36.3
CEGL007102		7	43.4
CEGL007136	Tsuga canadensis / Rhododendron maximum - (Clethra acuminata, Leucothoe fontanesiana) Forest	11	21.6
CEGL007230	Quercus alba - Quercus (rubra, prinus) / Rhododendron calendulaceum - Kalmia latifolia - (Gaylussacia ursina) Forest	46	55.1
CEGL007267	Quercus prinus - (Quercus rubra) - Carya spp. / Oxydendrum arboreum - Cornus florida Forest	32	56.1
CEGL007299	Quercus rubra / (Kalmia latifolia, Rhododendron maximum) / Galax urceolata Forest	27	35.8
CEGL007519		19	40.8
CEGL007693	Tsuga canadensis - Halesia tetraptera - (Fagus grandifolia, Magnolia fraseri) / Rhododendron maximum / Dryopteris intermedia Forest	29	43.3
CEGL007806	Fraxinus pennsylvanica - Ulmus americana / Carpinus caroliniana / Boehmeria cylindrica Forest	46	38.7
CEGL007842		13	65.9
CEGL008558		9	38.2

Model C

comm	scientific.name	nplots	rich
CEGL007470	Quercus falcata - Tilia americana var. caroliniana - Magnolia grandiflora / Ilex vomitoria Forest	5	57

Model D

comm	scientific.name	nplots	rich
CEGL007695	Aesculus flava - Acer saccharum - (Fraxinus americana, Tilia americana var. heterophylla) / Hydrophyllum canadense - Solidago flexicaulis Forest	45	72.2

No Model

comm	scientific.name	nplots	rich
CEGL003569		25	68.5
CEGL003595		11	99.3
CEGL003649		10	47.7
CEGL003658		7	63.3
CEGL003659		4	82.3
CEGL003661		17	58.2
CEGL004484	Nyssa biflora - (Nyssa aquatica, Taxodium distichum) Tidal Forest	6	39.3
CEGL004486		19	52.2
CEGL006124	Betula alleghaniensis / Ribes glandulosum / Polypodium appalachianum Forest	5	49.6
CEGL006137		5	46.8
CEGL006271	Quercus (prinus, coccinea) / Kalmia latifolia / (Galax urceolata, Gaultheria procumbens) Forest	41	36.0
CEGL007286		8	44.6
CEGL007431	Taxodium distichum - Nyssa aquatica / Fraxinus caroliniana Forest	22	27.2
CEGL007511		8	57.5
CEGL007543	Liriodendron tulipifera - Betula lenta - Tsuga canadensis / Rhododendron maximum Forest	42	49.4
CEGL007691	Quercus alba - Quercus coccinea - Quercus falcata / Kalmia latifolia - Vaccinium pallidum Forest	3	3
CEGL007710	Liriodendron tulipifera - Aesculus flava - (Fraxinus americana, Tilia americana) / Actaea racemosa - Laportea canadensis Forest	35	66.4
CEGL007711	Tilia americana var. heterophylla - Fraxinus americana - (Ulmus rubra) / Sanguinaria canadensis - (Aquilegia canadensis, Asplenium rhizophyllum) Forest	9	42.3
CEGL007719	Taxodium distichum - Fraxinus pennsylvanica - Quercus laurifolia / Acer rubrum / Saururus cernuus Forest	8	34.5
CEGL007730	Platanus occidentalis - Celtis laevigata - Fraxinus pennsylvanica / Lindera benzoin - Ilex decidua / Carex retroflexa Forest	10	45.7
CEGL007813	Juniperus virginiana var. silicicola - (Quercus virginiana, Sabal palmetto) Forest	5	31.8
CEGL00none		4	46.5

APPENDIX C: UNIVERSAL CODE

nicheutils.r

```
### Common functions useful for research programming
## Simple data structure manipulation

count <- function(x, use.na=FALSE) {
  ## counts the unique elements in a vector (levels if a factor)
  u <- if(is.factor(x)) levels(x) else unique(x)
  cnt <- sapply(u, function(i) sum(x==i, na.rm=TRUE))
  return(if(use.na) c(sum(is.na(x)), cnt) else cnt)
}

most.common <- function(x) { x[which.max(sapply(unique(x), function(i) sum(x==i)))] }

norm <- function(x) { (x - min(x, na.rm=TRUE)) / max(x - min(x, na.rm=TRUE), na.rm=TRUE) }

unwhich <- function(x, l=max(x)) { 1:l %in% x } ## turns a vector of element number
→s into logical index

replacematch <- function(x, patterns, no.match=NULL, ...) {
  # replaces each (string) element of x with the first pattern it matches
  # no.match = what to replace non-matched elements with (any string or NA); or not
→to (NULL)
  y <- as.character(x)
  for(i in patterns) y[grepl(pattern=i, x=y, ...)] <- i
  if(!is.null(no.match)) y[is.na(match(y, patterns))] <- no.match
  return(y)
}

## Graphing

linetypes <- c('solid', '12', '63', '4212', '921212', '3236')
linewidths <- c(1, 3, 1.5, 1.8, 1.5, 2.3)
contrast.colors <- function(n=10) {
  val <- switch(as.character(n), '1' = .65, '3' = c(.45, .85, .65), c(.45, .65))
  hsv(h=seq(0, 1, l=n+1 ) [1:n], v=val, s=1)
}

black.theme <- list(
  box.rectangle=list(col="black"),
  box.umbrella=list(col="black"),
  strip.background=list(col="white"),
  plot.symbol=list(col="black", pch=3, cex=.5)
)
```

simulate.r

```
### Code for generating species niches

rand.factorize <- function(x, groups=2) {
  ## aproximate random facotrization of any number, including primes

  d <- runif(groups)
  d <- groups * (d / sum(d))
  (x^(1/groups))^d
}
```

```

}

beta.abn <- function(gradientMax=100, alpha=1, gamma=1, peakPos=50, maxAbn=50, width=5
→0) {
  ## beta abundance generation using Minchin's additional parameters
  ## where peakPos is Minchin's m, maxAbn is his A0, and width is r

  gradient <- 1:gradientMax
  b <- alpha / (alpha + gamma)
  d <- b^alpha * (1 - b)^gamma

  abundances <- rep(0, length(gradient))
  is.present <- (peakPos - width * b) <= gradient & (peakPos + width * (1 - b)) >= g
→radient
  x <- gradient[ is.present ]
  abundances[ is.present ] <- (maxAbn / d) * ( ((x - peakPos) / width) + b)^alpha *
→(1 - ((x - peakPos) / width) + b))^gamma

  return(abundances)
}

sim.populate <- function(sites, draws, compet=1) {
  ## chooses species for a plot based on local pool
  ## sites is a matrix containing species (row) suitedness to each plot (col)

  if(length(draws)==1) draws <- rep(draws, dim(sites)[2])
  competition <- if(length(compet)==2) sample(compet[1]:compet[2], dim(sites)[2], TR
→UE)
  else rep(compet, dim(sites)[2])

  plots <- sapply( 1:dim(sites)[2], function(p) {
    pl <- sites[,p]
    if( sum(pl > 0) < 2) {
      if( sum(pl > 0) < 1) return( rep(0, length(pl)) )
      else return( as.numeric(pl > 0))
    }
    drawchance <- if(is.na(compet[1])) rep(1, length(pl))
    else replace(pl^competition[p], pl == 0, 0)
    observations <- sample( 1:length(pl), draws[p], replace=TRUE, prob=drawchance)
    counts <- sapply( 1:length(pl), function(sp) sum(observations == sp))
    relative <- counts / draws[p]
    return(relative)
  })
  dimnames(plots) <- dimnames(sites)
  return(plots)
}

make.plots <- function(niches, simPlots=500, plotBias=FALSE, quality=c(100,100), compe
→t=1) {
  ## create plots along resource gradients and populate based on niche curves
  nRes <- length(niches)
  gradientMax <- dim(niches[[1]])[2]
  nSpp <- dim(niches[[1]])[1]

  res <- t(matrix( if(!plotBias) { # amount of each r
→esource for the sites
    round( runif(nRes * simPlots, 1, gradientMax))
  } else {
    sample(1:gradientMax, nRes * simPlots, prob=(1 / exp(seq(2, 0 , l=gradientMax)
→), replace=TRUE )
  }, ncol=nRes, dimnames=list(paste("plot", 1:simPlots, sep=""), paste("r", 1:nRes ,

```

```

→ sep="")) ) )
  res <- res[, order(res[1,]), drop=FALSE] # order according to first resource va
→lue

  abundances <- sapply(1:simPlots, function(p) sapply(1:nSpp, function(sp) {
→# theoretical (maximum) abundance for each species at each site
    prod( sapply(nRes, function(r) niches[[r]][sp, res[r, p]] )) ^ (1 / nRes)
  )))
  dimnames(abundances) <- list( paste("sp", 1:nSpp, sep="") , paste("plot", 1:simPlo
→ts, sep="") )

  qualityGrad <- seq(quality[[1]], quality[[2]], length=dim(niches[[1]])[2])
→# site quality at each position on the gradient
  siteQuals <- colMeans( matrix( qualityGrad[ res ], ncol=dim(res)[2]) )

  plots <- sim.populate(abundances, siteQuals, compet)
  return(list(resources=res, plots=plots))
}

sim.niches <- function(resources=1, shape="normal", widthMethod="gamfun", meanWidth=25
→00,
  limWidth=TRUE, posMethod="uniform", numSpp=50, gradientMax=5000, tradeoff=FALSE, ...
→ ) {
  ## species curves for n niche-space dimensions.
  ## ... for make.plots

  gradient <- 1:gradientMax
  grandWidth <- pmax(round(switch(widthMethod,
    uniform = runif(numSpp, min=10, max=meanWidth * 2 ),
    gamfun = gradientMax * rgamma(numSpp, shape=.5, rate=1, scale=(2 * meanWidth /
→ gradientMax)),
    constant = meanWidth
  )), 10)

  widths <- matrix(round(gradientMax * t(sapply(
    grandWidth / gradientMax, rand.factorize, groups=resources ) )), ncol=resource
→s)

  spp.values <- lapply(1:resources, function(r) {
    peakPos <- switch(posMethod,
      biased = sample(gradient, numSpp, prob=(1 / exp(seq(2, 0, 1=gradientMax))
→), replace=TRUE ),
      uniform = round(runif(numSpp, 1, gradientMax)),
      equal = round(seq(1, gradientMax, length.out=numSpp))
    )
    width <- widths[,r]
    maxAbn <- if(tradeoff) 1 / width else rlnorm(numSpp, 2, 1)

    alpha <- switch(shape,
      random = runif(numSpp, .1, 4),
      normal = rep(1.99, numSpp),
      longtail = rep(20, numSpp),
      skewhalf = replace( rep(1.99, numSpp), sample(1:numSpp, numSpp / 2), .25),
→ # the skew method used in Fridley, where half are skewed left and half right
      skewrand = sample(c(.25, 1.99), numSpp, replace=TRUE) # actual
→random left/right skew
    )
    gamma <- switch(shape,
      random = runif(numSpp, .1, 4),
      normal = rep(1.99, numSpp),
      longtail = rep(20, numSpp),

```

```

        skewhalf = replace( rep(1.99, numSpp), alpha==1.99, .25), # mirror alpha
        skewrand = replace( rep(1.99, numSpp), alpha==1.99, .25)
    )
    return( data.frame(maxAbn, peakPos, width, alpha, gamma) )
})

niches <- lapply(spp.values, function(r) {
  t( sapply( 1:numSpp, function(i) {
    do.call("beta.abn", c(gradientMax, r[i,]))
  })))
})

width <- if(limWidth) {
  sapply(niches, function(r) rowSums(r > 0) ) / gradientMax
} else { sapply(spp.values, function(r) r$width ) / gradientMax }
height <- apply( sapply(spp.values, function(r) r$maxAbn ), 1, prod)
position <- apply( sapply(spp.values, function(r) r$peakPos ), 1, mean)

traits <- list(width=width, height=height, position=position)
plotdat <- make.plots(niches, ...)

compiled <- list(spp.traits=traits, niches=niches, plot.res=plotdat$resources, plo
→ts=plotdat$plots)
return(compiled)
}

```

params.r

```

### parameters for running tests and simulations

replicates <- 10

choices <- list(
  A = c( "none", "portion" ),
  C = c( 0, .2, .4 ),
  D = c( "n", "maxv", "cocombo" ),
  E = c( "n", "sub", "div", "codiv" )
)

## used by test.metrics and test.one.metric for calling sim.niches
simArgs <- list(
  resources = 1,
  gradientMax = 5000,
  numSpp = 50,
  shape = "normal",
  widthMethod = "gamfun",
  meanWidth = 2500,
  limWidth = FALSE,
  tradeoff = FALSE,
  posMethod = "uniform", # bias in position of species optima
  simPlots = 500,
  quality = c(20,20), # site draws bias
  plotBias = TRUE, # bias in plot resource
  compet = 1 # exponent controlling spp draws by height
)

## used by test.metrics and test.one.metric for resampling of a simulated dataset
metricArgs <- list(
  samples = 1, # how many times a set of possible sites should be res
→ampled

```

```

    plotsPerSample = 500,          # sample size
    minOccur = 1                  # values will not be calculated for species rarer than
→ this
)

```

APPENDIX D: CHAPTER 1 CODE

scenes.csv

scene	replicates	posMethod	plotBias	widthMethod	meanWidth
1	100	equal	FALSE	uniform	500
2	100	uniform	FALSE	uniform	500
3	100	biased	FALSE	uniform	500
4	100	equal	TRUE	uniform	500
5	100	uniform	TRUE	uniform	500
6	100	biased	TRUE	uniform	500
7	100	equal	FALSE	gamfun	500
8	100	uniform	FALSE	gamfun	500
9	100	biased	FALSE	gamfun	500
10	100	equal	TRUE	gamfun	500
11	100	uniform	TRUE	gamfun	500
12	100	biased	TRUE	gamfun	500
13	100	equal	FALSE	uniform	1000
14	100	uniform	FALSE	uniform	1000
15	100	biased	FALSE	uniform	1000
16	100	equal	TRUE	uniform	1000
17	100	uniform	TRUE	uniform	1000
18	100	biased	TRUE	uniform	1000
19	100	equal	FALSE	gamfun	1000
20	100	uniform	FALSE	gamfun	1000
21	100	biased	FALSE	gamfun	1000
22	100	equal	TRUE	gamfun	1000
23	100	uniform	TRUE	gamfun	1000
24	100	biased	TRUE	gamfun	1000
25	100	equal	FALSE	uniform	2500
26	100	uniform	FALSE	uniform	2500
27	100	biased	FALSE	uniform	2500
28	100	equal	TRUE	uniform	2500
29	100	uniform	TRUE	uniform	2500
30	100	biased	TRUE	uniform	2500
31	100	equal	FALSE	gamfun	2500
32	100	uniform	FALSE	gamfun	2500
33	100	biased	FALSE	gamfun	2500
34	100	equal	TRUE	gamfun	2500
35	100	uniform	TRUE	gamfun	2500
36	100	biased	TRUE	gamfun	2500
37	100	equal	FALSE	uniform	5000
38	100	uniform	FALSE	uniform	5000
39	100	biased	FALSE	uniform	5000
40	100	equal	TRUE	uniform	5000
41	100	uniform	TRUE	uniform	5000
42	100	biased	TRUE	uniform	5000
43	100	equal	FALSE	gamfun	5000
44	100	uniform	FALSE	gamfun	5000
45	100	biased	FALSE	gamfun	5000
46	100	equal	TRUE	gamfun	5000
47	100	uniform	TRUE	gamfun	5000
48	100	biased	TRUE	gamfun	5000

49	100	equal	FALSE	uniform	250
50	100	uniform	FALSE	uniform	250
51	100	biased	FALSE	uniform	250
52	100	equal	TRUE	uniform	250
53	100	uniform	TRUE	uniform	250
54	100	biased	TRUE	uniform	250
55	100	equal	FALSE	gamfun	250
56	100	uniform	FALSE	gamfun	250
57	100	biased	FALSE	gamfun	250
58	100	equal	TRUE	gamfun	250
59	100	uniform	TRUE	gamfun	250
60	100	biased	TRUE	gamfun	250
61	100	equal	FALSE	uniform	10000
62	100	uniform	FALSE	uniform	10000
63	100	biased	FALSE	uniform	10000
64	100	equal	TRUE	uniform	10000
65	100	uniform	TRUE	uniform	10000
66	100	biased	TRUE	uniform	10000
67	100	equal	FALSE	gamfun	10000
68	100	uniform	FALSE	gamfun	10000
69	100	biased	FALSE	gamfun	10000
70	100	equal	TRUE	gamfun	10000
71	100	uniform	TRUE	gamfun	10000
72	100	biased	TRUE	gamfun	10000

metrics.r

```
### Code for calculating niche width by various methods.
```

```
### choice A
```

```
normalize <- function(x, method="portion") {
  ## transform abundance table to make each sp's values relative to its own range or
  → mean
  if( method != "range" && method != "portion") stop("Invalid relativization method:
  → ", method)

  rel <- switch(method,
    portion = {
      scaled <- x / rowSums(x)
      scaled[is.na(scaled)] <- 0
      apply(scaled, 2, function(p) rowSums(replace(scaled, scaled > p, 0)) )
    },
    range = t( apply( x, 1, function(i) range = i / max(i) ))
  )
  return(rel)
}
```

```
### choice C
```

```
subset.plots <- function(x, method="quantile", value=.25) {
  subx <- switch( method,
    threshold = x >= value, # all occurrences equal
  → ing some cut-off
    quantile = {
      n.wanted <- ceiling(colSums(x > 0) * (1 - value))
      t(1 + t(dim(x)[1] - apply(x, 2, rank)) <= n.wanted)
    }
  )
}
```

```

    },
    top = apply(x, 2, function(p) {
      is.top <- (length(p) + 1) - rank(p, ties.method="max") <= value
      is.top[ p == 0] <- FALSE
      return(is.top)
    })
  )
  newx <- replace(x, !subx, 0)
  return(newx)
}

### choice D

calc.co.occur <- function(x, abundMethod="n", richAdj="n") {
  ## calculate co-occurrences separately for each spp (not combinations)
  ## takes plots for seeing co-occurrers, plots for focal spp, subcalculation method
  → (above), E flag

  values <- sapply( 1:dim(x)[1], function(f) {
    present <- x[f,] > 0
    if(sum(present) == 0) return(NA)
    fplots <- x[ , present, drop=FALSE]
    fplots[f,] <- 0
    frich <- colSums(fplots > 0)

    fvalues <- switch(abundMethod,
      "1" = 1, # test method
      "maxv" = apply(fplots, 1, max),
      "cocombo" = rowSums(unique(fplots > 0, MARGIN=2)),
      "combo" = {
        # expected <- sum(choose(mean(frich), 1:(mean(frich) - 1)))
        dim(unique(fplots > 0, MARGIN=2))[2] # / expected
      },
      rowSums(fplots) > 0
    )
    fadj <- sum(switch(richAdj,
      "sub" = sum(fvalues) - mean(frich),
      "div" = sum(fvalues) / (mean(frich) + 1),
      "codiv" = {
        co.rich <- rowSums( t(frich * t(fplots > 0)) ) / rowSums(fplots > 0)
        sum(fvalues / co.rich, na.rm=TRUE)
      },
      sum(fvalues)
    ), na.rm=TRUE)

    return(fadj)
  })
  names(values) <- dimnames(x)[[1]]
  return(values)
}

### Usable calculation methods

calc.multi.niche <- function(plots, choicelist=choices) {
  ## run all combinations of metric choices on an abundance table

  if( length( dim(plots)) != 2 ) stop( "input must be sp-plot abundance table")
  # if( min(plots) < 0 | max(plots) > 1 ) stop( "abundances must be relative")
  # if( dim(plots)[2] < 2) stop("some methods require multiple plots")

```

```

results <- lapply( choicelist$A, function(A) {
  A.result <- if( A=="none" ) plots else normalize(plots, A)
  lapply( choicelist$C, function(C) {
    C.result <- if(C == 0) A.result else subset.plots( A.result, "quantile", C
→ )
    lapply( choicelist$D, function(D) {
      lapply( choicelist$E, function(E) {
        DE.result <- calc.co.occure(x=C.result, abundMethod=D, richAdj=E)
        metric <- c(A, C, D, E)
        collect <- list( metric, DE.result )
        return(collect)
      })
    })
  })
  while( is.list(results[[1]])) results <- unlist(results, recursive=FALSE)      #
→ unlist, except last level
  results.metrics <- matrix(unlist( results[ c(TRUE, FALSE) ]), ncol=4, byrow=TRUE)
  results.values <- matrix(unlist( results[ c(FALSE, TRUE) ]), ncol=dim(plots)[1], b
→ yrow=TRUE)

  results.df <- data.frame(results.metrics, results.values)
  names(results.df) <- c( "A", "C", "D", "E", dimnames(plots)[[1]] )
  return(results.df)
}

resample.multi.niche <- function(plots, choicelist=choices, samples=100, plotsPerSampl
→ e=50, minOccur=20, allRep=FALSE) {
  ## wrapper for calc.multi.niche with resampling

  if(plotsPerSample > dim(plots)[2]) stop(paste("Not enough plots (", dim(plots)[2],
→ ") to meet sample size (", plotsPerSample, ").", sep=""))
  if(plotsPerSample < 2) stop("Some methods require multiple plots.")

  results <- lapply( 1:samples, function(i) {
    samp <- plots[, sample(1:dim(plots)[2], plotsPerSample)]
    r <- calc.multi.niche( plots=samp, choicelist=choicelist )
    common <- as.data.frame(append(list(A='com', C='com', D='com', E='com'), rowSu
→ ms(samp > 0) / plotsPerSample))
    return(rbind(r, common))
  })
  if(allRep) return(results)

  method.labels <- results[[1]][, 1:4]
  results.ave <- data.frame( lapply( 5:(4 + dim(plots)[1]), function(sp) {
    rowMeans( matrix( sapply(results, function(r) r[,sp] ), ncol=samples), na.rm=T
→ RUE )
  })))
  names(results.ave) <- row.names(plots)
  collected <- cbind(method.labels, results.ave)
  return(collected)
}

```

testmetrics.r

```

merge.df <- function(x) {
  ## utility: takes a list of similar data frames and combines them
  if(! is.list(x)) stop("input is not a list")
  merged <- data.frame(lapply( names(x[[1]]), function(cl) c( sapply(x, function(l)

```

```

→l[,cl] ))))
  names(merged) <- names(x[[1]])
  return(merged)
}

test.metrics <- function(replicates=100, simArgs, metricArgs, choicelist=choices, retu
→rnSim=FALSE) {
  ## tests all measures against simulated data

  results <- lapply(1:replicates, function(r) {
    message( paste("Doing replicate",r))

    true.widths <- NA
    while( sum(!is.na(true.widths)) < 3) {          # if sim unusable, redo
      sim <- do.call(sim.niches, simArgs)
      plots <- sim$plots
      too.rare <- rowSums(plots > 0) < metricArgs$minOccur
      true.widths <- replace( apply(sim$spp.traits$width, 1, prod), too.rare, NA
→)
      if(sum(!is.na(true.widths)) < 3) message("(Retrying)")
    }

    multi.niche.args <- append( list(plots=plots, choicelist=choicelist), metricAr
→gs)
    metrics <- cbind(r, do.call(resample.multi.niche, multi.niche.args))

    act.spp <- sum(rowSums(plots) > 0)
    richness <- mean(colSums(sim$plots > 0))
    overlap <- mean(colSums(sim$niches[[1]] > 0))
    saturation <- mean(colSums(sim$niches[[1]][, sim$plot.res] > 0) / colSums(sim$
→plots > 0), na.rm=TRUE)
    sim.prop <- list(richness, overlap, saturation, act.spp)
    names(sim.prop) <- c("richness","overlap","saturation","act.spp")

    result <- list(widths=true.widths, sim.prop=sim.prop, metrics=metrics, args=li
→st(simArgs, metricArgs))
    if(returnSim) result$sim <- sim
    return(result)
  })

  names(results) <- paste("r", 1:replicates, sep="")
  return(results)
}

```

runall

```

# submit LSF jobs for all metric-testing scenarios
# base parameters specified in params.r and variations in scenes.csv

MIN_SCENNUM=1
MAX_SCENNUM=72

for i in $(seq $MIN_SCENNUM $MAX_SCENNUM)
do
  bsub -q week R CMD BATCH --no-save --args -scennum=$i -- run.r run.Rout
done

```

run.r

```
### Simulate species, calculate niche breadths with all metrics, and compare to simulated values.
### Source from within R or run from command line: "R CMD BATCH run.r".

## setup

#!/usr/bin/Rscript

source("params.r")
source("metrics.r")
source("simulate.r")
source("testmetrics.r")

## set parameters

# get scenario number from command line option

scennum <- as.numeric(sapply(commandArgs(), strsplit, split="=")$"-scennum"[2])
if(is.null(scennum)) scennum <- 1

# load scenes file and change listed parameters
scenes <- read.table("scenes.csv", header=TRUE, sep="\t", stringsAsFactors=FALSE)
if(! scennum > dim(scenes)[1]) {
  replicates <- scenes[scennum, 2]
  for(r in 3:dim(scenes)[2]) {
    newval <- scenes[scennum, r]
    # newval <- eval(parse(text=as.character(scenes[scennum, r])))
    eval(substitute(simArgs$a <- newval, list(a = names(scenes)[r])))
  }
}

# set random seed
set.seed(scennum + as.numeric(substr(format(Sys.time(), "%X"), 7, 8)) / 100)

## execution

results <- test.metrics(replicates, simArgs=simArgs, metricArgs=metricArgs, choicelist
→=choices)

## output

corename <- paste("t", scennum, sep="")
i <- 1
while( file.exists( paste(corename, "_", i, ".Rdat", sep=""))) i <- i + 1
save( results, file=paste(corename, "_", i, ".Rdat", sep=""))
```

import.r

```
### Script to import data files and run basic analyses.
### Set home directory to data dir.

source("analysis.r")

## Read in the files

filenames <- list.files(".", pattern="t*Rdat")
scennums <- as.numeric(substr(sapply(filenames, function(i) { strsplit(i, "_", )[[1]][
```

```

→1] }),2,4))
file.n <- rle(scennums)$lengths
testdat <- lapply(1:length(file.n), function(i) {
  import.results( paste("t", unique(scennums)[i], sep=""), n=file.n[i] )
})
names(testdat) <- paste("t", unique(scennums), sep="")

## Calculate metric performance

perf <- lapply(testdat, FUN=lapply, calc.perform)
rm(testdat)

save(perf, file="intermediate.Rdat")

## Set parameters in output, clean up, and compress

scenes <- read.table("scenes.csv", header=TRUE, sep="\t", stringsAsFactors=FALSE)

# add scenario labels to rows
for(i in 1:length(perf)) { for(j in 1:length(perf[[1]])) {
  perf[[i]][[j]]$scennum <- as.numeric(substr(names(perf)[i], 2, 10))
}}

perf <- as.data.frame(merge.df(lapply(perf, merge.df)))
perf <- cbind(perf, scenes[perf$scennum, -(1:2)]) # add scenario parameters by
→label

perf$R2[ is.na(perf$R2) ] <- 0
perf$rank <- get.rank(perf)

perfmetric <- agg.by.metric(perf)

save.image(file="perf.Rwork")

```

analysis.r

```

library(doBy)
source("../nicheutils.r")

### Raw data input

import.results <- function(scenName, n=1) {
  ## reads results (multiple replicates) from files and combines into one df

  allresults <- list()
  for(i in 1:n) {
    load(paste(scenName, "_", i, ".Rdat", sep=""))
    allresults <- append(allresults, results)
  }
  for(i in 1:length(allresults)) allresults[[i]]$metrics$r <- i
  names(allresults) <- paste("r", 1:length(allresults), sep="")

  return(allresults)
}

calc.saturation <- function(x) {
  mean(colSums(x$niches[[1]][, x$plot.res] > 0) / colSums(x$plots > 0), na.rm=TRUE)
}

```

```

## Calculation of metric performance

calc.perform <- function(x) {
  n.metrics <- dim(x$metrics)[1]
  par.spp <- length(x$widths)
  actual <- na.omit(x$widths)
  if( length(actual)==0) return( rep(NA, n.metrics) )
  values <- x$metrics[, substr(names(x$metrics), 1, 2) == "sp"][, !is.na(x$widths),
→drop=FALSE]

  R2 <- apply(values, 1, function(m) {
    if( sum(!is.na(m)) == 0) return(0)
    summary( lm(actual ~ m) )$r.squared
  })

  act.width <- round(mean(actual), 2)
  co.occs <- round(rowMeans(values[1,], na.rm=TRUE), 2)

  collected <- data.frame(x$metrics[, substr(names(x$metrics), 1, 2) != "sp", m=1:n
→.metrics, R2,
    act.width, co.occs, lapply(x$sim.prop, rep, times=n.metrics),
    row.names=1:n.metrics)
  return(collected)
}

agg.by.metric <- function(x) {
  ## aggregate performance data into metric averages per scenario.

  scenvars <- paste(setdiff(names(x), c(
    "R2", "act.width", "act.spp", "richness", "overlap", "saturation", "rank", "r"
→, "co.occs"
  )), collapse=" + ")

  agg <- summaryBy(formula(paste(". ~ ", scenvars)), data=x, FUN=mean, keep.names=TR
→UE)
  agg$R2.sd <- summaryBy(formula(paste("R2 ~ ", scenvars)), data=x, FUN=sd)$R2
  agg$R2.min <- summaryBy(formula(paste("R2 ~ ", scenvars)), data=x, FUN=min)$R2
  agg$rankmin <- summaryBy(formula(paste("rank ~ ", scenvars)), data=x, FUN=min)$ran
→k
  return(agg[, -match("r", names(agg))])
}

## add a column to all results df, containing the mean R2 from scenario/replicate
# unsplit(scenes$mean.R2, list(all$par.width, all$r)) -> all$scenR2

find.mgroups <- function(x, firsts=FALSE) {
  # returns list vectors, grouping performance data row numbers into replicate group
→s.
  r <- 1:dim(x)[1]
  g <- split(r, list(x$shape, x$par.spp, x$par.width, x$m) ) <- 1:(dim(r) / 100)

  if(first) return( sapply(unique(g), function(i) which(g==i)[1] ) )
  else return(g)
}

### Analysis of performance

get.rank <- function(x) {

```

```

## average rank for each metric within scenarios (and replicates)
if(is.null(x$r)) x$r <- 1

ind <- as.list(x[, match( c(names(scenes)[-1:2]), "r"), names(x))])
ranks <- unlist(by(x$R2, ind, rank, ties.method="max"))
by.ids <- unlist(by(1:dim(x)[1], ind, I))
ranks.ord <- ranks[match(1:dim(x)[1], by.ids)]

return(ranks.ord)
}

get.rel <- function(x) {
  ## average rank for each metric within scenarios (and replicates)
  if(is.null(x$r)) x$r <- 1

  ind <- as.list(x[, match( c(names(scenes)[-1:2]), "r"), names(x))])
  ranks <- unlist(by(x$R2, ind, function(i) {
    i - mean(i)
  })))
  by.ids <- unlist(by(1:dim(x)[1], ind, I))
  ranks.ord <- ranks[match(1:dim(x)[1], by.ids)]
  return(ranks.ord)
}

find.Adiff <- function(x) {
  ## finds the difference between A-none and A-portion for the performance df of one
→ scenario
  spl <- split(x$R2, x$A)
  spl[[2]] - spl[[1]]
}

# Get typical plot richness for some scenario:
# mean(sapply(1:10, function(i) mean(colSums(sim.niches(numSpp=200)$plots > 0 )) ))

make.fake.rich <- function(draws=90) {
  spp <- seq(20, 200, 10)
  rich <- sapply( spp, function(i) {
    length( unique( sample(1:i, draws, TRUE, prob=runif(i))))
  })
  xa.frame(suited=spp, rich=rich)
}

```

graphmetrics.r

```

### Script to process metric results data and graph it
### Input is compiled perf and perfmtric dataframes from import.r
## Processing to include subsamp as a virtual option

library(lattice)
load("perf.Rdat")
load("perfmtric.Rdat")
source("../nicheutils.r")

perf$mm <- paste(perf$m, c("n","y")[(perf$subsamp + 1)], sep="")
perf <- transformBy(R2 ~ posMethod + widthMethod + meanWidth + plotBias + r,
  data=perf, rank=rank(R2) )

```

```

# Detect best metrics

best.mean <- rev( sort(c(by(perfmetric$rank, perfmetric$mm, mean))) )[1]
best.min <- rev( sort(c(by(perfmetric$rank, perfmetric$mm, min))) )[1]

## Graphing

ranklines <- c(1,seq(0,144,l=13)[-1])

plot.niches <- function(curves) {
  ## shows the surves generated by sim.niches

  colors <- rainbow(dim(curves)[1])
  plot(curves[1,], xlim=c(0,dim(curves)[2]), ylim=c(0,max(curves)), type="l")
  for(i in 2:dim(curves)[1]) {
    lines(curves[i,], col=colors[i])
  }
}

# Parameters and affects on all metrics

pdf("width.pdf")
xyplot(
  R2 ~ meanWidth/5000, group=widthMethod, data=perfmetric,
  panel=function(...) {
    panel.abline(h=seq(0,1,.1), col.line='#dddddd')
    panel.xyplot(..., col='black', type='a', lty=1:2, lwd=2)
  },
  auto.key=TRUE, ylim=c(-.05,1.05),
  main="Metric performance and parameters", xlab="mean niche width", ylab=expression(
→(r^2),
  key=list(
    lines=list(lty=1:2), col='black',
    text=list(c("gamma","uniform")) )
)
dev.off()

pdf("bias_position.pdf")
bwplot(
  R2 ~ as.factor(plotBias) | factor(posMethod,labels=c("biased","spaced","random")),
  data=perf,
  panel=function(...) {
    panel.abline(h=seq(0,1,.1), col.line='#dddddd')
    panel.bwplot(..., notch=TRUE, fill=c('#cccccc','#444444','#888888'))
  }, layout=c(3,1), par.settings=black.theme,
  scale=list(x=list(labels=c("random","biased"))),
  main="Site selection and niche position", xlab="site selection", ylab=expression(r
→^2)
)
dev.off()

# Comparison of metric choices

pdf("C_A_D.pdf")
xyplot(
  rank ~ C | factor(A, labels=c("unmodified","normalized")), group=as.factor(D), dat
→a=perfmetric,
  panel=function(...) {
    panel.abline(h=ranklines, col.line='#dddddd')

```

```

        panel.xyplot(..., type='a')
    },
    ylim=c(-7, 151), layout=c(2, 1), scale=list(y=list(at=c(1,48,96,144))),
    col='black', lty=linetypes[3:1], lwd=c(1, 1.5, 1), par.settings=black.theme,
    key=list(
        lines=list(lty=linetypes[3:1], lwd=c(1, 1.5, 1)),
        text=list(c("combinatoric", "max-value", "count")) ),
    main="Normalization, subsetting and counting method", xlab="subsetting level"
)
dev.off()

pdf("width_dist_D.pdf")
xyplot(
    rank ~ meanWidth/5000 | factor(widthMethod, labels=c("gamma distribution","uniform
→ distribution")),
    group=D, data=perfmtric,
    panel=function(...) {
        panel.abline(h=ranklines, col.line='#dddddd')
        panel.xyplot(..., type='a')
    },
    layout=c(2,1), ylim=c(-7, 151), scale=list(y=list(at=c(1,48,96,144))),
    lwd=c(1, 1.5, 1), lty=linetypes[3:1], col='black', par.settings=black.theme,
    key=list(
        lines=list(lty=linetypes[3:1], lwd=c(1,1.5,1)),
        text=list(c("combinatoric", "max-value","count")) ),
    main="Niche width and distribution, and counting method", xlab="mean niche width"
)
dev.off()

subsampdiff <- subset(perfmtric, subsamp==TRUE)$rank - subset(perfmtric, subsamp==FA
→LSE)$rank
pdf("subsamp_bias_A.pdf")
bwplot(
    subsampdiff ~ A | factor(plotBias, labels=c("random sampling","biased sampling")),
    data=subset(perf, subsamp==TRUE),
    panel=function(...) {
        panel.abline(h=seq(-72, 72, 12)[-7], col.line='#dddddd')
        panel.abline(a=0)
        panel.bwplot(..., fill=c('#888888', '#cccccc'), notch=1)
    },
    scales=list(x=list(labels=c("no","yes")), y=list(at=c(-72, -36, 0, 26, 72) )),
    layout=c(2,1), auto.key=TRUE, ylim=c(-72, 72), par.settings=black.theme,
    main="Performance, niche position and plot position", xlab="normalization", ylab="
→rank change with subsampling "
)
dev.off()

pdf("satur_E.pdf")
xyplot(
    rank ~ saturation, group=E, data=perf,
    panel=function(...) {
        panel.abline(h=ranklines, col.line='#dddddd')
        panel.xyplot(..., type='smooth', lty=linetypes[c(2:3,1,4)], col='black', lwd=c
→(1.5, 1, 1, 1))
    },
    ylim=c(-7,151), scale=list(y=list(at=c(1,48,96,144))), par.settings=black.theme,
    key=list(
        lines=list(lty=linetypes[c(2, 3, 1, 4)], lwd=c(1.5, 1, 1, 1)),
        text=list(c("co-division","division","none","subtraction")) ),
    main="Richness correction methods and saturation"
)

```

```

dev.off()

# Top-performing metrics

pdf("best_width.pdf")
xyplot(
  rank + rankmin ~ meanWidth/5000, groups=mm,
  data=subset(perfmetric, mm %in% c("1n", "2y", "55n")),
  panel=panel.superpose,
  panel.groups=function(..., col.line, lwd=1, lty, type, group.number) {
    panel.abline(h=ranklines, col.line='#ddddd')
    panel.average(..., col.line='black', type='l', horiz=FALSE, lty=linetypes[grou
→p.number],
                                lwd=c(1, 1.5, 1)[group.number], fun=if(panel.number()==1) mean e
→lse min)
  },
  ylim=c(-7, 151), scale=list(y=list(at=c(1,48,96,144))), par.settings=black.theme,
  key=list(
    lines=list(lty=linetypes[1:3], lwd=c(1, 1.5, 1)),
    text=list(c("1n", "2y", "55n")) ),
  main="Selected metric comparison", xlab="mean niche width", ylab="rank",
  strip=strip.custom(factor.levels=c("mean rank", "minimum rank")))
)
dev.off()

pdf("best.pdf")
bwplot(
  R2 ~ mm, data=subset(perf, mm %in% c('1n','2y','55n')),
  panel=function(...) {
    panel.abline(h=seq(0,1,.1), col.line="#ddddd")
    panel.bwplot(..., notch=TRUE, fill='grey')
  },
  par.settings=black.theme, box.ratio=2,
  main="performance of selected metrics", xlab="metric", ylab=expression(r^2)
)
dev.off()

# Graphing CVS species widths

pdf("CVScommon.pdf")
xyplot(
  t55n ~ common, data=widths.rel, pch=20, cex=.5, col='black',
  main="commonness and theta of CVS species", xlab="relative commonness", ylab="rela
→tive theta (55n)"
)
dev.off()

```

APPENDIX E: CHAPTER 2 CODE

spatialSim.r

```

metric <- list(A="portion", C=0.2, D="maxv", E="div")

makeSpace <- function(side=50, scales=c(10,3,1), compress=F) {
  ## generates grid with smoothed z-values between 0 and 1

```

```

## done for each specified scale, which are output as the z-dim in an array, or co
→mopressed
## reps and wgts can also be vectors, thus varying with each grain size (in all re
→sources)

plots <- array(sapply(scales, function(s) {
  if(s > .8 * side) warning("Scale is almost as large as area; distortion likely
→")
  pl <- GaussRF(x=1:side,y=1:side, model="stable", grid=T, param=c(1, 1, 0, max(
→s, .00000001), 2))
  pl <- (pl - min(pl)) * (1 / max(pl - min(pl)))
}), dim=c(side, side, length(scales)))
if(compress) plots <- array(apply(plots, c(1, 2), mean), dim=c(side, side, 1))
return(plots)
}

findBlocks <- function(side, grain) {
  lblocks <- floor( side / grain)
  edge <- side - (lblocks * grain)
  blocks <- c(sapply(1:lblocks, function(i) {
    rep(c(rep(1:lblocks + lblocks * (i - 1), each=grain), rep(0,edge)), grain)
  }), rep(0, edge * side))
  return(blocks)
}

spatialSim <- function(draws=1, blockGrain=c(1,2,4,8), totSampSize=Inf,
  ssSize=50, fixArea=FALSE, nss=100, spaceArgs=list(), simArgs=list()) {
  ## makes spacial plots and runs theta metric on them at multiple grain sizes
  ## fixArea varies the number of plots in a subsample to maintain a constant area (
→= ssSize, which otherwise is n plots)

  space <- do.call(makeSpace, spaceArgs)
  resources <- if(spaceArgs$compress) mean(spaceArgs$scales) else spaceArgs$scales
  nres <- length(resources)
  simArgs$resources <- nres

  spp <- do.call(sim.niches, simArgs)
  nspp <- dim(spp$spp.traits$width)[1]
  widths <- apply(spp$spp.traits$width, 1, prod)
  position <- spp$spp.traits$position

  suited.res <- sapply(1:nres, function(r) { spp$niches[[r]][, space[,r] * 4999 + 1
→]
  })
  suited <- array(apply(suited.res, 1, prod), dim=c(nspp, length(space[,1])))
→
  # suitedness using all resources
  dimnames(suited) <- list(paste("sp", 1:dim(suited)[1], sep=""), paste("site", 1
→:dim(suited)[2], sep=""))
  sites <- sim.populate(suited, rep(draws, dim(suited)[2]))
→
  # populated grid cells for use in blocks/plots

  results <- merge.df(lapply(blockGrain, function(g) {
    blocks <- findBlocks(side=sqrt(dim(sites)[2]), grain=g) # what block is
→ each site in?
    nPlots <- if(fixArea) round(totSampSize / g^2) else totSampSize # plots
→ in total sample
    sampled.blocks <- if(max(blocks) <= nPlots) 1:max(blocks) else sample(1:max(bl
→ocks), nPlots, FALSE)
    block.plots <- sapply(sampled.blocks, function(b) { # what
→spp. are present in each block?
      rowMeans(sites[, blocks==b, drop=FALSE] )
    })
  })

```

```

blocks.per.ss <- if(fixArea) { max(2, round(ssSize / g^2)) }
else { min(max(blocks), ssSize) }
nss <- if(blocks.per.ss == max(blocks)) 1 else nss

theta.raw <- sapply(resample.multi.niche(plots=block.plots, choicelist=metric,
  plotsPerSample=blocks.per.ss, samples=nss, minOccur=1, allRep=TRUE),
  unlist)[- (1:4), , drop=FALSE]
theta <- rowMeans(theta.raw, na.rm=TRUE)
theta.var <- apply(theta.raw, 1, var, na.rm=TRUE)

rich <- mean(colSums(block.plots > 0))
hetero <- sapply(1:nres, function(r) {
  rang <- sapply(sampled.blocks, function(b) range(space[,r][blocks==b] ))
  mean(rang[2,] - rang[1,])
})
g.results <- data.frame(
  row.names=1:nspp,
  grain=g, rich, hetero, resource=do.call(paste, append(list(sep="_"), resou
→rces)),
  sp=1:nspp, theta, theta.var, width=c(widths), position
)
return(g.results)
}))
return(results)
}

```

graintest.r

```

### All steps to analyze grain and theta

## Initialization

rm(list=ls())
library(lattice)
library(RandomFields)
library(doBy)
library(reshape)
source("../nicheutils.r")
source("spatialSim.r")
source("grain_analysis.r")
source("../simulate.r")
source("../metrics/metrics.r")

rainb2 <- hcl( # dark purple to yellow (more continuous lightness)
  h=seq(350, 70,l=100),
  l=seq(0,100,l=100),
  c=c( seq(0,130,l=50), seq(130,95,l=50) ))

spatArgs = list(
  blockGrain = c(2,3,5,7,10,14), draws=1, totSampSize=5000, ssSize=1000, fixArea=TRU
→E, nss=1,
  simArgs = list(numSpp=50, meanWidth=2000, limWidth=FALSE, compet=1),
  spaceArgs = list(side=150, scales=c(1), compress=F)
)
par.space <- c(1,3,10,30,100)
par.meanWidth <- c(500,1000,5000) # paired with nspp so mean spp density on niche a
→xis = 10
par.nspp <- c(100,50,10)

```

```

## Produce some images to illustrate space-generation

# spacel <- makeSpace(150, 1, F)[,,1]
# spacel0 <- makeSpace(150, 10, F)[,,1]
# spacel00 <- makeSpace(150, 100, F)[,,1]
# space.mix <- spacel + spacel0 + spacel00
# pdf("spacel.pdf"); levelplot(spacel, col.regions=rainb2, main="Smoothing of 1"); dev
→.off()
# pdf("spacel0.pdf"); levelplot(spacel0, col.regions=rainb2, main="Smoothing of 10");
→dev.off()
# pdf("spacel00.pdf"); levelplot(spacel00, col.regions=rainb2, main="Smoothing of 100"
→); dev.off()
# pdf("space_mix.pdf"); levelplot(space.mix, col.regions=rainb2, main="Smoothing of 1,
→ 10 & 100"); dev.off()

## Run simulations with difference resource scales and niche overlaps

spatsraw <- lapply(1:(length(par.space) * length(par.meanWidth)), function(i) {
  print(i)

  spatArgs$spaceArgs$scales <- par.space[ceiling(i / length(par.meanWidth))]
  spatArgs$simArgs$meanWidth <- par.meanWidth[(i - 1) %% length(par.meanWidth) + 1]
  spatArgs$simArgs$numSpp <- par.nspp[(i - 1) %% length(par.meanWidth) + 1]

  spat <- repSpatialSim(spatArgs, 100)

  spat$results$par.width <- spatArgs$simArgs$meanWidth
  spat$perform$par.width <- spatArgs$simArgs$meanWidth
  return(spat)
})
spats <- list( results = merge.df(lapply(spatsraw, function(i) i$results)) )
spats$perform <- merge.df(lapply(spatsraw, function(i) i$perform))
rm(spatsraw)

spats$results$resource <- as.integer(as.character(spats$results$resource))
spats$perform$resource <- as.integer(as.character(spats$perform$resource))

spats$results <- transformBy(
  ~ resource + rep + grain, data=spats$results,
  rel.mean=I(theta / mean(theta, na.rm=TRUE)),
  rel.linear=norm(theta),
  rel.ecdf=ecdf(theta)(theta)
)

spats$perform$gamma <- summaryBy(
  theta ~ resource + rep + grain, spats$results, FUN=function(i) sum(!is.na(i)) )[,
→4]

# per-species variability between grains

persp.var <- summaryBy(
  data=spats$results,
  theta + rel.mean + rel.linear + rel.ecdf ~ resource + rep + sp,
  FUN = function(i) var(i, na.rm=TRUE), keep.names=TRUE
)

# find the single best grain for a scenario

spats$bestgrain <- summaryBy(r2 ~ resource + rep, spats$perform, FUN=function(i) {

```

```

        weighted.mean(spatArgs$blockGrain, i) })
names(spats$bestgrain)[4] <- "weighted.best"

spats$bestgrain$weighted.best <- summaryBy(
  r2 ~ resource + rep, data=spats$perform, FUN=function(i) {
    weighted.mean(unique(spats$perform$grain^2), i)
  })[,3]

# summary of best grains

bestgrains <- summaryBy(weighted.best^2 ~ resource + par.width, data=spats$bestgrain,
  FUN=function(i) c(quantile(i, p=.25), quantile(i, .5), quantile(i, p=.75) ))
names(bestgrains)[3:5] <- c("lo", "mean", "hi")

save.image(file="spats.Rwork")

## Graph simulation results

pdf("theta.pdf")
xyplot(
  theta ~ grain^2, data=subset(spats$results, !is.na(theta)), group=resource,
  ylim=c(0,3),
  type='a', lty=linetypes[1:5], lwd=linewidths[1:5], col=contrast.colors(5),
  main="Generalism scores and grain", xlab="grain area", ylab="mean spp. generalism"
→,
  key=list(
    title="landscape homogeneity", cex.title=1,
    text=list(as.character(c(1,3,10,30,100))),
    lines=list(lty=linetypes[1:5], lwd=linewidths[1:5], col=contrast.colors(5)) )
)
dev.off()

pdf("bias.pdf")
xyplot(
  theta + rel.mean + rel.linear + rel.ecdf ~ resource, data=persp.var,
  panel=panel.superpose, panel.groups=panel.average, fun=function(i) mean(i, na.rm=T
→RUE),
  horiz=F, type='l', lty=linetypes[1:4], lwd=linewidths[1:4], col=contrast.colors(4)
→,
  main="Grain bias correction", xlab="landscape homogeneity", ylab="niche estimate v
→ariation",
  ylim=c(-.05, .55),
  key=list(
    text=list(c('none', 'around mean', 'linear', 'ECDF')),
    lines=list(lty=linetypes[1:4], lwd=linewidths[1:4], col=contrast.colors(4))
  )
)
dev.off()

pdf("perform.pdf")
xyplot(r2 ~ grain^2, groups=resource, data=spats$perform,
  type='a', lty=linetypes[1:5], lwd=linewidths[1:5], col=contrast.colors(5),
  main="Metric performance and grain", xlab="grain area", ylab=expression(r^2),
  key=list(
    title="landscape homogeneity", cex.title=1,
    text=list(as.character(c(1,3,10,30,100))),
    lines=list(lty=linetypes[1:5], lwd=linewidths[1:5], col=contrast.colors(5))

```

```

    )
  )
dev.off()

pdf("hetero.pdf")
xyplot(r2 ~ hetero, data=spats$perform,
  panel=function(...) {
    panel.xyplot(..., type='p', pch=20, cex=.5, col=contrast.colors(1))
    panel.loess(..., lwd=2, col='black')
  },
  ylim=c(-.05, 1.05),
  main="Metric performance and heterogeneity", xlab="mean plot resource range", ylab
→=expression(r^2)
)
dev.off()

pdf("hetcause.pdf")
xyplot(hetero ~ grain^2, groups=resource, data=spats$perform,
  type=c('a','p'), pch='|', cex=1, ylim=c(-.05, 1.05), lty=linetypes[1:5], lwd=linewidth
→idths[1:5], col=contrast.colors(5),
  key=list(
    title="resource smoothing", cex.title=1,
    text=list(as.character(c(1,3,10,30,100))),
    lines=list(lty=linetypes[1:5], lwd=linewidths[1:5], col=contrast.colors(5))
  ),
  main="Causes of heterogeneity", xlab="grain area", ylab="plot heterogeneity"
)
dev.off()

pdf("var_best.pdf")

densityplot(
  spats$bestgrain$max.var.diff,
  col='black', pch='|', main="Grain selection by maximum variation", xlab="differenc
→e in area"
)
dev.off()

pdf("bestgrain.pdf")
xyplot( I(weighted.best^2) ~ resource, data=spats$bestgrain,
  panel=function(x,y) {
    r <- summaryBy(weighted.best ~ resource, spats$bestgrain, FUN=c(min, max))
    grid.polygon(
      x=c(r$resource, rev(r$resource)),
      y=c(r$weighted.best$max^2, rev(r$weighted.best$min^2)),
      gp=gpar(fill='grey', col=NA), default.units='native'
    )
    panel.average(x, y, horiz=F, col='black', fun=mean, lwd=2)
  },
  main="Best grain", ylab="grain area", xlab="landscape homogeneity",
  ylim=c(-5, 105),
  key=list(lines=list(lwd=c(5,1), col=c('grey','black')), text=list(c("range","mean"
→)))
)
dev.off()

```

grainCVS.r

```

### Import CVS and USDA data with nests for multi-scale spatial analysis
## save 'taxon importance' data into CSV

```

```

library(reshape)
library(doby)
library(lattice)
library(grid)
source("../nicheutils.r")

## import

plants <- read.table("plants.csv", TRUE, sep=",", fill=TRUE, encoding="UTF-8")
CVS <- read.table("taxa.csv", header=TRUE, sep=",", fill=TRUE, encoding="UTF-8")

names(CVS)[1] <- "name"
CVS$name <- factor(sub(" var\\. .*", "", CVSmodules$name))
corner.cols <- grep("corner", names(CVS))
names(CVS)[corner.cols] <- c("corner1", "corner4", "corner5", "corner3", "corner2")
CVS[, corner.cols] <- replace(CVS[, corner.cols], is.na(CVS[, corner.cols]), 0)

NCcounties <- read.table("NCcounty.csv", header=TRUE, sep=",", fill=TRUE, encoding="UTF-8")

# pare down to necessary data

CVSmodules <- subset(CVS, ! Stratum.Index %in% c("--all--", "C", "E", "H", "S", "U", "mod R"
→))
CVSmodules <- summaryBy(
  corner1 + corner2 + corner3 + corner4 + corner5 ~ name + Stratum.Index + OBSERVATI
→ON_ID,
  data=CVSmodules, id=~sppFAMILY + Author.Observation.Code, keep.names=TRUE, FUN=min
)

# collect summaries

spp.names <- unique(CVSmodules$name)
spp.families <- CVSmodules[match(spp.names, CVSmodules$name), "sppFAMILY"]
common.families <- unique(spp.families)[which(apply(unique(spp.families), function(f)
→ sum(spp.families==f, na.rm=TRUE)) > 50)]
nspp <- length(spp.names)
lvls <- data.frame(
  length = c(37336, 100, 10, 3.16, 1, .32, .1),
  area = c(1.394e+9, 1000, 100, 10, 1, .1, .01),
  name = c('lv1C', 'lv15', 'lv14', 'lv13', 'lv12', 'lv11', 'lv10') )

## separate into presence matrices by sampling level

counties <- t(apply(strsplit(gsub(" ", "", as.character(NCcounties[, "County"])), "(
→),")", function(i) NCcounty.names %in% i ))
rownames(counties) <- NCcounties$Scientific.Name
colnames(counties) <- NCcounty.names

plots <- sapply(by(
  CVS[CVS$Stratum.Index=="--all--", ],
  INDICES=list(CVS[CVS$Stratum.Index=="--all--", "OBSERVATION_ID"]),
  FUN=function(i) { spp.names %in% i$name }
), unlist)

nests <- by(CVSmodules,
  INDICES=list(CVSmodules$OBSERVATION_ID, as.character(CVSmodules$Stratum.Index)),
  FUN=function(i) {

```

```

        blank <- matrix(0, ncol=5, nrow=nspp)
        blank[match(i$name, spp.names),] <- as.matrix(i[grepl("corner", names(CVSmodule
→s))])
        if(is.null(blank)) warning()
        rownames(blank) <- spp.names
        colnames(blank) <- c("corner1", "corner4", "corner5", "corner3", "corner2")

        return(blank)
    }
)
nests[unlist(lapply(nests, function(i) is.null(i[[1]]))) <- NULL # remove nonexistent
→ combinations

modules <- sapply(nests, rowSums) > 0
submodules <- matrix(unlist(nests), nrow=length(spp.names)) # most be subset to des
→ ired level

## run width calculations

source("theta55n.r")

lv1C <- theta.55n(counties)
lv10 <- theta.55n(plots)
lv11 <- theta.55n(modules)
lv12 <- theta.55n(submodules >= 2)
lv13 <- theta.55n(submodules >= 3)
lv14 <- theta.55n(submodules >= 4)
lv15 <- theta.55n(submodules >= 5)

thetas <- data.frame(lv1C, lv10, lv11, lv12, lv13, lv14, lv15, name=spp.names, spfam=s
→ pp.families)
thetas[,1:7] <- replace(thetas[,1:7], is.na(thetas[,1:7]), 0)
thetas$mean <- rowMeans(thetas[,1:7])

family.means <- sapply(levels(spp.families), function(i) mean(subset(thetas, spfam==i
→ $mean)))
thetas.rel <- cbind( apply(thetas[,1:7], 2, function(i) ecdf(i)(i) ), thetas[, 8:dim(t
→ hetas)[2]] )
thetas.rel$min <- lvls$name[apply(thetas.rel[,1:7], 1, which.min)]

## package for analysis

thrk <- melt(thetas, id.vars=c("name"), measure.vars=c(2:7, 1), variable_name="lv1")
thrk$rel <- melt(thetas.rel, id.vars=c("name"), measure.vars=c(2:7, 1), variable_name=
→ "lv1")$value

thrk$area <- lvls[match(thrk$lv1, lvls$name), "area"]
thrk$habit <- plants[match(thrk$name, plants$Scientific.Name, NA), "Growth.Habit"]
habitfreq <- (count(thrk$habit)/7)[-1]
thrk$native <- (!unwhich(grep("I", plants$Native.Status), 56673))[match(thrk$name, pla
→ nts$Scientific.Name, NA)]
thrk$common.plot <- rep(rowSums(plots > 0), 7)

spptrends <- data.frame(
    name = thetas$name,
    spfam = thetas$spfam,
    common = rowSums(plots > 0),
    maxlv1 = apply(thetas.rel[,1:7], 1, function(i) lvls$area[which.max(i)] ),

```

```

minlvl = apply(thetas.rel[,1:7], 1, function(i) lvl$area[which.min(i)] ),
sd = apply(thetas.rel[,1:7], 1, function(i) sd(i) )
)

save.image(file="CVSgrain.Rwork")

### graph it

lvl$scale <- list( x=list(at=c(-2, -1, 0, 1, 2, 3, 9)))

pdf("CVS_levels.pdf")
xyplot(
  value ~ log(area, 10), data=thr,
  panel=function(x,y) {
    r <- summaryBy(value ~ area, thr, FUN=c(min, max))
    grid.polygon(
      x=log(c(r$area, rev(r$area)), 10),
      y=c(r[,3], rev(r[,2])),
      gp=gpar(fill='grey', col=NA), default.units='native'
    )
    panel.average(x, y, horiz=F, col='black', fun=mean, lwd=2)
  },
  scale=lvl$scale, ylim=c(-8, 158), xlim=c(-2.55, 9.55),
  xlab=expression(log[10]~sampling~area~(m^2)), ylab="species generalism", main="sam
→ pling scale and generalism",
  key=list(lines=list(lwd=c(5,1), col=c('grey','black')), text=list(c("range","mean"
→ )))
)
dev.off()

pdf("CVS_levels_rel.pdf")
xyplot(
  rel ~ log(area, 10), data=thr,
  panel=function(x,y) {
    r <- summaryBy(rel ~ area, thr, FUN=c(min, max))
    grid.polygon(
      x=log(c(r$area, rev(r$area)), 10),
      y=c(r[,3], rev(r[,2])),
      gp=gpar(fill='grey', col=NA), default.units='native'
    )
    panel.average(x, y, horiz=F, col='black', fun=mean, lwd=2)
  },
  scale=lvl$scale, ylim=c(-.05, 1.05), xlim=c(-2.55, 9.55),
  xlab=expression(log[10]~sampling~area~(m^2)), ylab="normalized species generalism"
→ , main="sampling scale and generalism",
  key=list(lines=list(lwd=c(5,1), col=c('grey','black')), text=list(c("range","mean"
→ )))
)
dev.off()

pdf("CVS_mg_hist.pdf")
histogram(
  log(spptrends$min, 10), breaks=10, col='grey', border=0,
  xlim=c(-3, 11), scale=list(x=list(at=seq(-2, 10, 2))),
  xlab="grain where niche estimate is minimized"
)
dev.off()

```

```

pdf("CVS_traits.pdf")
# Species generalism across scales, by native status and growth habit as defined by US
→DA;
# the 4 most common habits are included (n=1450, 73% of total). 252 species (10%) were
→ non-native.
xyplot(
  rel ~ log(area, 10) | as.character(habit), groups= c("native","non-native")[native
→ + 1],
  data=thrk[(habitfreq >= 90)[match(thrk$habit, names(habitfreq))], ],
  panel=panel.superpose, panel.groups=panel.average, FUN=mean, horiz=F, type='l',
  par.settings=append(
    simpleTheme(lty=linetypes[1:2], lwd=linewidths[1:2], col=contrast.colors(2)),
    list(strip.background=list(col="white"))
  ),
  layout=c(4,1),
  auto.key=list(points=FALSE, lines=TRUE, cex.title=1),
  scale=list(
    x=list(at=c(-2, -1, 0, 1, 2, 3, 9), labels=c('-2',' ',' ',' ',' ','3','9')),
    y=list(at=seq(0, 1, .25))
  ),
  xlab=expression(log[10]~sampling~area~(m^2)), ylab="relative generalism",
  main="sampling scale, growth habit and native status"
)
dev.off()

pdf("CVS_select_spp.pdf")
xyplot(
  rel ~ log(area, 10), groups=as.character(name),
  data=subset(thrk, name %in% c(
    "Quercus geminata", "Gratiola neglecta", "Mikania scandens", "Rhynchospora car
→eyana")),
  type='a',
  par.settings=simpleTheme(lty=linetypes[1:4], lwd=linewidths[1:4], col=contrast.col
→ors(4)),
  auto.key=list(points=FALSE, lines=TRUE), ylim=c(-.05, 1.05), scale=lvlscale,
  xlab=expression(log[10]~sampling~area~(m^2)), ylab="relative generalism",
  main="sampling scale and selected plant species"
)
dev.off()

```

grain_analysis.r

```

analyze.spat.width <- function(x) {
  ## takes output from spatialSim and computes r-squared values for simulated / calc
→ulated widths
  ## (i.e. 'width' and 'theta' columns) for each grain and resource

  if(is.null(x$rep)) x$rep <- 1
  byindex <- list(x$resource, x$grain, x$rep)

  perf <- aggregate(1:dim(x)[1], byindex, function(i) {
    summary(lm(width ~ theta, data=x[i,]))$r.squared
  })
  names(perf) <- c("resource", "grain", "rep", "r2")

  mean.theta <- aggregate(x$theta, byindex, mean, na.rm=TRUE)$x
  mean.theta.var <- aggregate(x$theta.var, byindex, mean, na.rm=TRUE)$x
  props <- x[c(by(1:dim(x)[1], byindex, function(i) i[1] )), c("hetero","rich")]
  return(cbind(perf, props, mean.theta, mean.theta.var))
}

```

```

repSpatialSim <- function(arglist=spatArgs, reps=10) {
  z <- replicate(reps, do.call(spatialSim, arglist), FALSE)
  for(i in 1:reps) z[[i]]$rep <- i
  spats <- merge.df(z)
  spats.lm <- analyze.spat.width(spats)
  return(list(results=spats, perform=spats.lm))
}

find.theta.mean <- function(x) {
  stats <- aggregate(x$theta, list(x$grain, x$rep, x$resource), mean, na.rm=TRUE )
  names(stats) <- c("grain", "rep", "resource", "theta.mean")
  stats$theta.var <- aggregate(x$theta, list(x$grain, x$rep, x$resource), var, na.rm
→=TRUE )$x
  return(stats)
}

plot.spat.results <- function(x, dots=FALSE) {
  xyplot(
    theta ~ width | grain, groups=resource, data=x,
    layout=c(length(unique(x$grain)), 1),
    auto.key=T, cex=.1, pch=1:3, type=c('smooth', if(dots) 'p')
  )
}

plot.spat.perform <- function(x) {
  xyplot(r2 ~ grain^2, groups=resource, data=x,
    type=c('p', 'a'), pch=1:length(unique(x$resource)), cex=.5,
    ylim=c(0,1),
    auto.key=T, xlab="grain area", ylab=expression(r^2),
    main="Metric performance and grain"
  )
}

```

APPENDIX F: CHAPTER 3 CODE

CVSmanip.r

```

### Code to import CVS taxa data and covnvert to matrix
# (make sure csv files are loaded without initial garbage)

source("../theta55n.r")
source("../nicheutils.r")
library(doBy)

plots <- subset(read.table("../CVS/AllCVS_plot_data.csv", TRUE, ",", fill=TRUE), Area=
→1000)
names(plots)[1] <- "Author.Observation.Code"
taxa <- subset(read.table("../CVS/AllCVS_taxa.csv", TRUE, ",", fill=TRUE, encoding="UT
→F-8"), authorObsCode %in% plots$Author.Observation.Code)
names(taxa)[1] <- "name"

taxa$name <- sub(" $| var\\. .*", "", taxa$name)
taxa.simple <- summaryBy(cover ~ name, taxa, id=~sppFamily + OBSERVATION_ID + authorOb
→sCode, FUN=sum)
plotids <- unique(taxa.simple$authorObsCode)

```

```

plots <- plots[match(plotids, plots$Author.Observation.Code),]

CVS <- t(sapply(by(taxa.simple, list(taxa.simple$name), FUN=function(i) {
  replace(rep(NA, length(plotids)), match(i$authorObsCode, plotids, 0), i$cover)
}), unlist))
colnames(CVS) <- plotids
CVS <- replace(CVS, is.na(CVS), 0)

CVSwidths <- theta.55n(CVS)

save.image(file="CVS.Rwork")

```

CVS_pth.r

```

### Script to analyze CVS data re: spp. niche widths and plot generalism

library(geneplotter)
library(lattice)
library(reshape)
library(doBy)
source("plot_theta.r")
source("CVSmanip.r")
choices <- list(A="portion", C=0.2, D="maxv", E="div")

## plot calculations

comm <- read.table("../CVS/AllCVS_comm.csv", TRUE, ",", fill=TRUE)

CVSpth <- summ.plots.theta(CVS, CVSwidths, TRUE)
CVSpth$project <- as.numeric(substr(colnames(CVS), 1,3))
CVSpth$comm <- comm[match(rownames(CVSpth), comm$Author.Observation.Code), "community.c
ode"]
#CVSpth <- summ.spp.theta(CVS, CVSwidths)

## physiognomic type summaries

communities <- aggregate(CVSpth[, -which(names(CVSpth)=="comm")], list(CVSpth$comm), me
an)
names(communities)[1] <- "comm"
communities$nplots <- sapply(communities$comm, function(i) sum(CVSpth$comm == i, na.rm
=TRUE) )

communities$mean.lm <- c(by(CVSpth, factor(CVSpth$comm), FUN=function(i) {
  if(dim(i)[1] < 3 | sum(is.na(i$theta.mean))==dim(i)[1]) return(NA)
  m <- summary(lm(theta.mean ~ norm(rich), i))$coefficients["norm(rich)", ]
  if(is.na(m["Pr(>|t|)"]) | m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
}))
communities$min.lm <- c(by(CVSpth, factor(CVSpth$comm), FUN=function(i) {
  if(dim(i)[1] < 3 | sum(is.na(i$theta.min))==dim(i)[1]) return(NA)
  m <- summary(lm(theta.min ~ norm(rich), i))$coefficients["norm(rich)", ]
  if(is.na(m["Pr(>|t|)"]) | m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
}))
communities$max.lm <- c(by(CVSpth, factor(CVSpth$comm), FUN=function(i) {
  if(dim(i)[1] < 3 | sum(is.na(i$theta.max))==dim(i)[1]) return(NA)
  m <- summary(lm(theta.max ~ norm(rich), i))$coefficients["norm(rich)", ]
  if(is.na(m["Pr(>|t|)"]) | m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
}))
communities$sd.lm <- c(by(CVSpth, factor(CVSpth$comm), FUN=function(i) {

```

```

    if(dim(i)[1] < 3 | sum(is.na(i$theta.sd))==dim(i)[1]) return(NA)
    m <- summary(lm(theta.sd ~ norm(rich), i))$coefficients["norm(rich)", ]
    if(is.na(m["Pr(>|t|)"]) | m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
  })

## Matching of models to communities

communities$model <- "none"
attach(communities)
communities[which(mean.lm > 0 & sd.lm >= 0 & min.lm > 0 & max.lm > 0), "model"] <- "C"
communities[which(mean.lm >= 0 & sd.lm > 0 & min.lm < 0 & max.lm > 0), "model"] <- "D"
communities[which(mean.lm < 0 & abs(sd.lm) < .05 & min.lm < 0 & max.lm >= 0), "model"]
→ <- "B"
communities[which(abs(mean.lm) < .05 & abs(sd.lm) < .05 & min.lm >= 0 & max.lm >= 0),
→ "model"] <- "A"
communities[nplots < 3, "model"] <- "NA"
detach()

communities$scientific.name <- as.character(comm[match(communities$comm, comm$communit
→ y.code), "scientific.name"])

# find what communities each species is present in / vice-versa
spp.comm <- sapply(unique(CVSpth$comm), function(pr) {
  rowSums(CVS[, which(CVSpth$comm==pr), drop=FALSE]) > 0
})
names(spp.comm) <- unique(CVSpth$comm); rownames(spp.comm) <- rownames(CVS)

## Graph

pdf("CVS_pth.pdf", 4, 8)
plot.pth(CVSpth, main="CVS plot diversity and generalism")
dev.off()

pdf("comm_hist.pdf", 4, 8)
histogram(
  ~ value | variable,
  data=melt(communities, "rich", measure.vars=c("mean.lm", "sd.lm", "min.lm", "max.lm"))
→ ,
  breaks=do.breaks(c(-1.11, 1.11), 21),
  layout=c(1,4), index.cond=list(c(3,4,2,1)),

  col= replace(rep("black",21), 11, "grey"),
  key=list(text=list("n = 232"), x=.75,y=.95 ),
  xlab="dependence of generalism on richness",
  main="generalism within CVS projects",
  strip=FALSE, strip.left=strip.custom(
    factor.levels=c("mean", "standard deviation", "minimum", "maximum"),
    bg='white'),
)
dev.off()

pdf("comm_lines.pdf")
xyplot(theta.mean ~ rich, group=comm,
  data=subset(CVSpth, comm %in% communities$comm[which(communities$nplots > 40)]),
  type='smooth'
)
dev.off()

```

```
## Output community matches

for(i in c("A", "B", "C", "D", "none", "NA")) {
  write.table(
    subset(communities, model==i)[, c("comm","scientific.name","nplots","rich")],
    file=paste(i, "comm.csv", sep=""),
    sep=";", row.names=FALSE, col.names=TRUE, quote=TRUE
  )
}
```

plot_theta.r

```
### Functions for connecting plots with species thetas

library(vegan)
library(reshape)
library(geneplotter)
library(lattice)
source("../nicheutils.r")

summ.plots.theta <- function(plots, thetas, useRank=TRUE) {
  ## takes abundance table and spp niche widths, and summarizes plots in a dataframe
  # useRank normalizes raw thetas to 0-1

  rich <- colSums(plots > 0)
  theta.plots <- (if(useRank) norm(thetas) else unlist(thetas)) * (plots > 0)
  theta.mean <- colSums(theta.plots, na.rm=TRUE) / rich
  theta.sd <- apply(theta.plots, 2, function(p) sd(p[p!=0], na.rm=TRUE))
  theta.min <- apply(theta.plots, 2, function(p) {
    if(sum(p, na.rm=TRUE) == 0) 0 else min(p[p!=0], na.rm=TRUE)
  })
  theta.max <- apply(theta.plots, 2, max, na.rm=TRUE)
  shannon <- diversity(plots, MARGIN=2)
  simpson <- diversity(plots, MARGIN=2, index="simpson")

  summ <- data.frame(rich, theta.mean, theta.sd, theta.min, theta.max, shannon, simp
→son)
  rownames(summ) <- colnames(plots)
  return(summ)
}

summ.spp.theta <- function(x, thetas, useRank=TRUE) {
  theta.plots <- (if(useRank) {
    norm(thetas) }
    else { unlist(thetas) }) * (x > 0)
  spth <- as.data.frame(t(sapply(1:dim(x)[1], function(sp) {
    co.theta <- theta.plots[-sp, x[sp, , drop=FALSE] > 0, drop=FALSE]
    if(dim(co.theta)[2] == 0) return(c(sp, 0, 0, 0, 0, 1, 0))
    theta.mean <- mean(co.theta[co.theta > 0], na.rm=TRUE)
    theta.sd <- sd(co.theta[co.theta > 0], na.rm=TRUE)
    theta.min <- min(co.theta[co.theta > 0], na.rm=TRUE)
    theta.max <- max(co.theta[co.theta > 0], na.rm=TRUE)
    rich.mean <- mean(colSums(co.theta > 0, na.rm=TRUE)) + 1
    rich.sd <- sd(colSums(co.theta > 0, na.rm=TRUE))
    return(c(sp, theta.mean, theta.sd, theta.min, theta.max, rich.mean, rich.sd))
  })))
  names(spth) <- c("sp", "theta.mean", "theta.sd", "theta.min", "theta.max", "rich.m
→ean", "rich.sd")
}
```

```

    spth$theta <- if(useRank) rank(thetas, na.last='keep') / length(thetas) else theta
→s
    return(spth)
}

summ.spp.plot.theta <- function(x, xpth, xthetas) {
  ## for each spp finds the mean properties of the plots it occurs in
  pl <- as.data.frame(t(sapply(1:dim(x)[1], function(sp) {
    occur <- unlist(x[sp,] > 0)
    colMeans(xpth[occur, sapply(xpth[1,], is.numeric) ], na.rm=TRUE)
  })))
  pl$width <- xthetas
  return(pl)
}

## graphing

plot.pth <- function(x, xvar="rich", xlab="richnes", main="plot richness and generalis
→m", ...) {
  if(is.null(x$project)) x$project <- 0
  xyplot(
    formula(paste("value ~", xvar, "| variable")),
    data=melt(x, measure.vars=c("theta.mean", "theta.sd", "theta.min", "theta.max")),
    panel=function(x,y, ...) {
      if(sd(y, na.rm=TRUE) > .02 & length(unique(y)) > 5) {
        panel.smoothScatter(x,y, ..., nrpoints=0, colramp=colorRampPalette(c('
→white', '#888888')) )
      } else {
        panel.xyplot(x,y, ..., col='#888888', pch=20, cex=.33)
      }
      if(length(unique(x)) > 25) {
        panel.loess(x,y, ..., col='black', lwd=1)
      } else {
        panel.average(x,y, ..., col='black', lwd=1, horiz=FALSE)
      }
    },
    strip=FALSE, strip.left=strip.custom(
      factor.levels=c("mean", "standard deviation", "minimum", "maximum"),
      bg='white'),
    scales=list(y=list(at=c(0,.5,1), alternating=c(0,1,0,1)) ),
    index.cond=list(c(3,4,2,1)), layout=c(1,4), aspect=1,
    xlab=xlab, ylab="plot generalism", main=main, ...
  )
}

```

null_pth.r

```

### Simulate plot data and analyze re: richness and generalism

## setup

source("plot_theta.r")
source("../simulate.r")
source("../theta55n.r")
source("../params.r")
library(latticeExtra)

baseArgs <- list(
  resources=1, tradeoff=FALSE, compet=0, limWidth=FALSE, quality= c(50, 50), simPlot
→s=100,

```

```

    meanWidth=2500, widthMethod='uniform', numSpp=50)
nrep <- 10

## generate scenarios

simA <- replicate(nrep, do.call(sim.niches, replace(baseArgs,
  c("compet", "quality"), list(NA, c(5, 100)) )))
# simA.thetas <- sapply(simA["plots",], theta.55n)
simA.thetas <- sapply(simA["spp.traits",], function(i) i$width )
simA.pth <- merge.df(lapply(1:dim(simA)[2], function(i) summ.plots.theta( simA[,i]$plo
→ts, simA.thetas[,i]) ))
simA.pth$rep <- rep(1:nrep, each=baseArgs$simPlots)
simA.pth$rich <- unlist(by(simA.pth$rich, list(simA.pth$rep), norm))

simB <- replicate(nrep, do.call(sim.niches, baseArgs))
simB.thetas <- sapply(simB["spp.traits",], function(i) i$width )
simB.pth <- merge.df(lapply(1:dim(simB)[2], function(i) summ.plots.theta( simB[,i]$plo
→ts, simB.thetas[,i]) ))
simB.pth$rep <- rep(1:nrep, each=baseArgs$simPlots)
simB.pth$rich <- unlist(by(simB.pth$rich, list(simB.pth$rep), norm))

simC <- replicate(nrep, do.call(sim.niches, replace(baseArgs,
  c("tradeoff", "compet"), list(TRUE, 5) )))
simC.thetas <- sapply(simC["spp.traits",], function(i) i$width )
simC.pth <- merge.df(lapply(1:dim(simC)[2], function(i) summ.plots.theta( simC[,i]$plo
→ts, simC.thetas[,i]) ))
simC.pth$rep <- rep(1:nrep, each=baseArgs$simPlots)
simC.pth$rich <- unlist(by(simC.pth$rich, list(simC.pth$rep), norm))

simD <- replicate(nrep, do.call(sim.niches, replace(baseArgs,
  c("tradeoff", "compet"), list(TRUE, c(-10, 10)) )))
simD.thetas <- sapply(simD["spp.traits",], function(i) i$width )
simD.pth <- merge.df(lapply(1:dim(simD)[2], function(i) summ.plots.theta( simD[,i]$plo
→ts, simD.thetas[,i]) ))
simD.pth$rep <- rep(1:nrep, each=baseArgs$simPlots)
simD.pth$rich <- unlist(by(simD.pth$rich, list(simD.pth$rep), norm))

all.pth <- rbind(simA.pth, simB.pth, simC.pth, simD.pth)
all.pth$model <- rep(c("A", "B", "C", "D"), each=dim(simA.pth)[1])

## make graphs

pdf("sims.pdf")
useOuterStrips(
  xyplot(
    value ~ rich | model * variable,
    data=melt(all.pth, measure.vars=c("theta.mean", "theta.sd", "theta.min", "theta.m
→ax"))),
  panel=function(x,y, ...) {
    if(sd(y, na.rm=TRUE) > .02) panel.smoothScatter(
      x,y, ..., nrpoints=0, colramp=colorRampPalette(c('white', '#888888'))
    )
    panel.loess(x,y, ..., col='black', lwd=1)
  },
  scales=list(
    y=list(draw=FALSE), x=list(draw=FALSE) ),
  index.cond=list(1:4, c(3,4,2,1)), layout=c(4,4), aspect=1,
  xlab="richness", ylab="plot generalism", main="simulation models"
),

```

```

strip=function(..., bg) strip.default(..., bg='white'),
strip.left=strip.custom(
  factor.levels=c("mean", "standard deviation", "minimum", "maximum"), bg='white
→')
)
dev.off()

## Analyze for slopes

simA.melt <- melt(simA.pth, c("rich", "rep"), measure.vars=c("theta.mean","theta.sd","
→theta.min","theta.max"))
Alm <- by(simA.melt, list(simA.melt$rep, simA.melt$variable), FUN=function(i) {
  m <- summary(lm(value ~ rich, i))$coefficients["rich",]
  if(m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
})

simB.melt <- melt(simB.pth, c("rich", "rep"), measure.vars=c("theta.mean","theta.sd","
→theta.min","theta.max"))
Blm <- by(simB.melt, list(simB.melt$rep, simB.melt$variable), FUN=function(i) {
  m <- summary(lm(value ~ rich, i))$coefficients["rich",]
  if(m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
})

simC.melt <- melt(simC.pth, c("rich", "rep"), measure.vars=c("theta.mean","theta.sd","
→theta.min","theta.max"))
Clm <- by(simC.melt, list(simC.melt$rep, simC.melt$variable), FUN=function(i) {
  m <- summary(lm(value ~ rich, i))$coefficients["rich",]
  if(m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
})

simD.melt <- melt(simD.pth, c("rich", "rep"), measure.vars=c("theta.mean","theta.sd","
→theta.min","theta.max"))
Dlm <- by(simD.melt, list(simD.melt$rep, simD.melt$variable), FUN=function(i) {
  m <- summary(lm(value ~ rich, i))$coefficients["rich",]
  if(m["Pr(>|t|)"] >= .1) 0 else m["Estimate"]
})

```

text_pth.r

```

### Extracting character data from a text and converting to abundance table
## file is preprocesses to remove non-letter characters and convert to lowercase
## NB: ensure source file has no trailing blank line

source("../plot_theta.r")
source("../metrics/metrics.r")
source("../theta55n.r")

scan(file="ch1.txt", what="character", blank.lines.skip=FALSE) -> ch1

blanks <- c(0, which(ch1==""), length(ch1))
par.ind <- lapply(2:length(blanks), function(i) (blanks[i-1] +1):(blanks[i]-1))
par.words <- lapply(par.ind, function(i) ch1[i] )

abund <- t(sapply(unique(ch1), function(w) sapply(par.words, function(p) sum(p==w) )))
rel.abund <- t(t(abund) / colSums(abund))

## Analysis, as for any abundance table

```

```

thetas <- theta.55n(rel.abund)
plot.theta <- summ.plots.theta(rel.abund, thetas)

pdf("theta_hist.pdf")
histogram(thetas, xlab="word generalism")
dev.off()

pdf("plot_theta.pdf", 4, 8)
plot.pth(plot.theta, main="Darwin text sentence richness and generalism")
dev.off()

# pdf("sp_theta.pdf")
# plot.pth(sp.theta, main="Word location generalism")
# dev.off()

pdf("theta_common.pdf")
xyplot(thetas ~ rowSums(rel.abund > 0), panel=function(...) {
  panel.xyplot(..., type='p', col='black', pch=20, cex=.75)
  panel.loess(..., span=1, degree=2)
},
  xlab="n presences", ylab="generalism"
)
dev.off()

```

domesday_pth.r

```

###
## Import and prep

source("../plot_theta.r")
source("../../metrics/metrics.r")
source("../../theta55n.r")

read.table("domesdaystatistics_livestock.tab", header=TRUE, sep="\t", fill=NA) -> lives
→stock
stock <- livestock[,c(5,7,9,11,13,15,17,19)]
stock$horses.1066 <- replace(livestock$other.1066, ! livestock$other.codes.1066 %in% c
→("horses", "horses at the hall", "foal", "horse", "mares"), 0)
stock$horses.1066 <- replace(livestock$other.1066, ! livestock$other.codes.1066 %in% c
→("ox", "oxen"), 0)
stock <- replace(stock, is.na(stock), 0)
rownames(stock) <- livestock$structidx

## calculations

stock.rel <- stock / pmax(1, rowSums(stock, na.rm=TRUE))
thetas <- theta.55n(stock.rel)
names(thetas) <- unlist(strsplit(names(thetas), ".1066"))
stock.pth <- summ.plots.theta(stock.rel, thetas, TRUE)

## graph

pdf("livestock_1066.pdf", 4, 8)
plot.pth(stock.pth, title="Livestock and manor generalism in England, 1066", subset=ri
→ch>0)
dev.off()

## 1086 data

```

```

stock.1086 <- livestock[,c(4,6,8,10,12,14,16,18)]
stock.1086$horses.1086 <- replace(livestock$other.1086, ! livestock$other.codes.1086 %
→in% c("horses","horses at the hall","foal","horse","mares"), 0)
stock.1086$horses.1086 <- replace(livestock$other.1086, ! livestock$other.codes.1086 %
→in% c("ox","oxen"), 0)
stock.1086 <- replace(stock.1086, is.na(stock.1086), 0)
rownames(stock.1086) <- livestock$structidx

stock.1086.rel <- t(stock.1086 / pmax(1, rowSums(stock.1086, na.rm=TRUE)))
thetas.1086 <- theta.55n(stock.1086.rel)
names(thetas.1086) <- unlist(strsplit(names(thetas.1086), ".1086"))
stock.1086.pth <- summ.plots.theta(stock.1086.rel, thetas.1086, FALSE)

pdf("livestock_1086.pdf", 4, 8)
plot.pth(stock.1086.pth, title="Livestock and manor generalism in England, 1086")
dev.off()

## combine

rbind(stock.pth, stock.1086.pth) -> stock.all.pth
plot.pth(stock.all.pth,
  title="Livestock and manor generalism in England", group=rep(c(1066,1086), each=32
→30))

pdf("theta_change.pdf")
xyplot(thetas.1086 ~ thetas, ylim=c(.3,2), xlim=c(.3,2),
  panel=function(...) {
    panel.xyplot(..., pch=20,col='black',cex=2)
    panel.abline(0,b=1, lty=2)
    panel.text(..., names(thetas), pos=1)
  },
  main="Change in species generalism", xlab="Generalism, 1066", ylab="Generalism, 10
→86"
)
dev.off()

xyplot(
  theta.mean ~ rich, data=stock.all.pth,
  group=rep(c(1066,1086),each=3230), type=c('p','a'), auto.key=TRUE, pch=c(3,4)
)

```