Predicting global community properties from uncertain estimates of interaction strengths

Supporting Information

György Barabás & Stefano Allesina

Abstract

This Supporting Information presents a step-by-step description of our methodology for generating random interaction matrices and Allometric Trophic Networks (Section 1), shows further results obtained from our simulations (Section 2), and gives extra details on the theoretical background of matrix binning (Section 3). We also describe a simple algorithm for obtaining the ε-pseudospectral contour lines of any given matrix (Section 4). Finally, we derive a new metric for assessing the degree of nonnormality of a matrix, which is useful in assessing how sensitive its spectrum is expected to be to perturbing its entries (Section 5).

Contents

1 More details on the simulation methods

1.1 Random interaction matrices

Here is a step-by-step breakdown of how we created our random interaction matrices.

1. We first pick a species richness S and a connectance C (i.e., the fraction of nonzero interactions in the matrix). In our simulations S was either 50, 100, 250, or 500, and C was chosen from 0.1, 0.25, 0.5, or 1.

2. Consider the following procedure for generating a probability distribution. First pick a distribution shape from two options: either lognormal or Gamma. Then determine the given distribution's mean by sampling it uniformly from $[0.1,\ 10]$. Finally, the standard deviation is also sampled uniformly, from $[1,\ 10]$.

3. Use this procedure to generate four separate probability distributions. Call the first one "PredPrey" for predator-prey, the second one "Mut" for mutualism, the third "Comp" for competition, and the last one "Diag" for diagonal.

4. How do we decide whether to pick a lognormal or a Gamma distribution? We actually repeated every simulation with all 16 possible combinations of the distributions, with "PredPrey", "Mut", "Comp", and "Diag" all taking on both possible values, so nothing was left out.

5. Generate two random numbers, both of them sampled from $[0,\ 1]$. Call them C_t and C_m (for "trophic" and "mutualistic").

6. Now generate an $S \times S$ matrix A of all zeros.

7. Make a fraction $C_t C$ of the offdiagonal entries predator-prey, a fraction $C_m C$ mutualistic, and a fraction $C(1 - C_t - C_m)$ competitive (leave the rest of the $1 - C$ entries as zeros). The predator-prey entries are drawn from "PredPrey", the mutualistic ones from "Mut", and the competitive ones from "Comp". For predator-prey interactions, make sure that the (i, j)th and (j, i)th entries of A have the opposite sign; for mutualism, both should be positive, and for competition, both negative.

8. Finally, the diagonal of the matrix is generated: fill out the diagonal entries by sampling from the distribution "Diag". Multiply the diagonal by -1 to make its entries negative.

9. Note: one could introduce a "diagonal connectance" C_d, i.e., the fraction of nonzero diagonal entries. However, in our simulations we always set this to 1, so no diagonal entries were left equal to zero.

Here is a function, written in R (R Development Core Team 2008), that we used to implement this procedure:

```
# This function generates a random interaction matrix with a prescribed
# proportion of trophic, mutualistic, and competitive interactions.
#
# Input
# -----
# S: number of species.
# C: connectance (fraction of nonzero interactions); between 0 and 1.
# ppred: probability that a nonzero link is predator-prey (between 0 and 1).
# pcomp: probability that a nonzero non-predator-prey link is competitive
#        (between 0 and 1); the rest are mutualistic.
# conv: conversion efficiency of predators; realistically between 0 and 1.
# pdist, mdist, cdist: distribution from which trophic, mutualistic, and
#        competitive entries are drawn. Values: 1 = lognormal; 2 = Gamma.
# mp, sp, mm, sm, mc, sc: sample means and standard deviations for each
#        distribution. The (mp, sp) are mean and std dev for predator-prey,
#        (mm, sm) are for mutualism, and (mc, sc) for competition.
# Cd: fraction of diagonal entries that are nonzero.
# ddist: distribution from which nonzero diagonal entries are drawn.
#        Values: 1 = lognormal; 2 = Gamma. Note: the diagonal entries are
#        never positive (i.e., the values drawn from the chosen distribution
#        get multiplied by -1).
# md, sd: sample mean and standard deviation of nonzero diagonal entries.
#
# Output
# ------
# An S-by-S matrix
#
GenerateMatrix <- function(S, C, ppred, pcomp, conv, pdist, mdist, cdist,
                           mp, sp, mm, sm, mc, sc, Cd, ddist, md, sd) {
    # Generate adjacency matrix. Only upper triangle is generated; the
    # diagonal and lower triangular part are discarded. Entry is 0 for
    # absence and 1 for presence of interaction.
    A <- matrix(sample(0:1, S*S, replace=TRUE, prob=c(1-C, C)), nrow=S, ncol=S)
    A[lower.tri(A)] <- 0
    diag(A) <- 0
    # Now set trophic (= 1) vs nontrophic (= 2) interactions:
    A <- A * matrix(sample(1:2, S*S, replace=TRUE, prob=c(ppred, 1-ppred)),
                    nrow=S, ncol=S)

    # Create predator-prey adjacency matrix AP
    AP <- A
    AP[AP!=1] <- 0 # discard nontrophic (= 2) interactions
    AP <- AP * matrix(sign(rnorm(S*S)), nrow=S, ncol=S) # either -1 or 1
    AP <- AP - t(AP)
```

```
# Create mutualistic and competitive adjacency matrices AM and AC
AM <- A
AM[AM!=2] <- 0
AM[AM==2] <- 1
AM <- AM * matrix(sample(c(-1, 1), S*S, replace=TRUE,
                         prob=c(pcomp, 1-pcomp)), nrow=S, ncol=S)
AM <- AM + t(AM)
AC <- AM
AM[AM==(-1)] <- 0 # mutualism
AC[AC==1] <- 0 # competition

# Predator-prey values
AP[AP==1] <- conv * AP[AP==1] # conversion efficiencies
if (pdist==1) {
    p1 <- log(mp) - log(1 + sp^2/mp^2)/2
    p2 <- sqrt(log(1 + sp^2/mp^2))
    AP <- AP * matrix(rlnorm(S*S, p1, p2), nrow=S, ncol=S)
}
if (pdist==2) {
    p1 <- mp^2 / sp^2
    p2 <- sp^2 / mp
    AP <- AP * matrix(rgamma(S*S, shape=p1, scale=p2), nrow=S, ncol=S)
}

# Mutualism values
if (mdist==1) {
    p1 <- log(mm) - log(1 + sm^2/mm^2)/2
    p2 <- sqrt(log(1 + sm^2/mm^2))
    AM <- AM * matrix(rlnorm(S*S, p1, p2), nrow=S, ncol=S)
}
if (mdist==2) {
    p1 <- mm^2 / sm^2
    p2 <- sm^2 / mm
    AM <- AM * matrix(rgamma(S*S, shape=p1, scale=p2), nrow=S, ncol=S)
}

# Competition values
if (cdist==1) {
    p1 <- log(mc) - log(1 + sc^2/mc^2)/2
    p2 <- sqrt(log(1 + sc^2/mc^2))
    AC <- AC * matrix(rlnorm(S*S, p1, p2), nrow=S, ncol=S)
}
if (cdist==2) {
    p1 <- mc^2 / sc^2
    p2 <- sc^2 / mc
    AC <- AC * matrix(rgamma(S*S, shape=p1, scale=p2), nrow=S, ncol=S)
```

```
    }

    # Finally, create the diagonal of the matrix
    diagvec <- sample(0:1, S, replace=TRUE, prob=c(1-Cd, Cd))
    if (ddist==1) {
        p1 <- log(md) - log(1 + sd^2/md^2)/2
        p2 <- sqrt(log(1 + sd^2/md^2))
        diagvec <- diagvec * rlnorm(S, p1, p2)
    }
    if (ddist==2) {
        p1 <- md^2 / sd^2
        p2 <- sd^2 / md
        diagvec <- diagvec * rgamma(S, shape=p1, scale=p2)
    }

    # The sum of all components is the full interaction matrix
    return(AP + AM + AC - diag(diagvec))
}
```

1.2 The Allometric Trophic Network model

To generate community matrices using the Allometric Trophic Network model, we followed the equations and parameterization described by Berlow et al. (2009). First, a food web's adjacency matrix w_{ij} is created using the niche model (Williams and Martinez 2000; see below), where w_{ij} is equal to 1 if species i eats species j and to 0 otherwise. The dynamical equations of the model read

$$\frac{\mathrm{d}B_i}{\mathrm{d}t} = r_i G_i(N_1,N_2) B_i - x_i B_i + \sum_{j=1}^{S} w_{ij} x_i y B_i F_{ij} - \sum_{j=1}^{S} w_{ji} x_j y B_j F_{ji}/e_{ji}, \tag{1}$$

where B_i is the biomass of species i, r_i is a nutrient uptake-dependent maximum growth rate, $G_i(N_1,N_2)$ is the growth achieved on the two nutrients N_1 and N_2, x_i is a metabolic rate, y is the maximum consumption rate of consumers relative to their metabolic rate, e_{ij} is species i's assimilation efficiency when eating species j, and F_{ij} is a generalized functional response given by

$$F_{ij} = \frac{\omega_{ij} B_j^h}{B_0^h + cB_i B_0^h + \sum_{k=1}^{S} w_{ik}\omega_{ik} B_k^h}. \tag{2}$$

In this functional response, ω_{ij} is the proportion of i's maximum consumption rate targeted at consuming j, B_0 is a half-saturation density, h is an exponent determining the Holling type of the functional response, and c is a predator interference parameter.

We assume there are two limiting nutrients N_1 and N_2 which the primary producers can consume. Their dynamics is given by

$$\frac{\mathrm{d}N_i}{\mathrm{d}t} = D(s_i - N_i) - \sum_{j=1}^{S} c_{ij} r_i G_i(N_1, N_2) B_i, \tag{3}$$

where N_i is the concentration of nutrient i, D is a turnover rate, c_{ij} is the content of nutrient i in the biomass of species j, s_i is nutrient i's supply rate, and $G_i(N_1, N_2)$ is the same function that appeared in Eq. (1). It is defined by

$$G_i(N_1, N_2) = \min\left(\frac{N_1}{K_{1i} + N_1},\ \frac{N_2}{K_{2i} + N_2}\right), \tag{4}$$

where "min" picks the smaller of the two arguments, and K_{ij} is species j's half saturation density for nutrient i.

The parameters were assigned as follows.

1. $S = 50$ for the initial number of species.
2. Out of these 50, the number of basal species (primary producers) was sampled uniformly as an integer from $[2,\ 10]$. In our ordering of species, they are the last ones, so if there are two producers, they will be species 49 and 50.
3. The food web adjacency matrix w_{ij} (equal to 1 if species i eats species j and to 0 otherwise) is generated by the niche model (Williams and Martinez 2000), in the following way.
   - Primary producers only consume nutrients, therefore $w_{ij} = 0$ for all i that are primary producers.
   - Each consumer species i consumes a range of other species $j, j+1, \ldots, j+k$, where the starting index j is uniformly sampled between $i+1$ and S, and the length k is an integer sampled from $[-(S-i+1)/3,\ (S-i+1)/3]$.
   - Note: if $j+k$ turns out to be larger than S, it is set equal to S, and if it turns out less than $i+1$, it is set to $i+1$.
4. $\omega_{ij} = 1/\sum_{j=1}^{S} w_{ij}$ if the sum in the denominator is nonzero; otherwise, $\omega_{ij} = 0$.
5. $r_i = 1$ for species i that are primary producers and 0 otherwise.
6. Determine the trophic level T of each species. Trophic level can be calculated by defining the matrix $A_{ij} = w_{ij}/\sum_{k=1}^{S} w_{ik}$ and the vector $u_i = 1$ (for all i); we then have $T_i = \sum_{j=1}^{S} (I-A)^{-1}_{ij} u_j$, where I is the $S \times S$ identity matrix and $(I-A)^{-1}_{ij}$ is the (i,j)th entry of the inverse of the matrix $I - A$.

7. Define a vector Z of predator-prey body mass ratios; its S entries are sampled from a lognormal distribution with mean 10 and variance 100.

8. Define the vector of body masses. For predators, the body masses are given by the formula $M_i = Z_i^{T_i - 1}$. For species that are not predators, the body masses are equal to 1.

9. Using this, the metabolic rates x_i are calculated as $x_i = (a_x/a_r)\, M_i^{-0.25}$, where M_i is species i's body mass, and (a_x/a_r) is equal to 0.138 for primary producers (i.e., species with trophic level exactly equal to one), and to 0.314 otherwise.

10. Assimilation efficiencies: $e_{ij} = 0.45$ if the trophic level of species i is exactly 2; otherwise, $e_{ij} = 0.85$.

11. The K_{ij} are uniformly and independently sampled from $[0.1,\ 0.2]$.

12. $c_{1i} = 1$ and $c_{2i} = 0.5$ for all i, $y = 8$, $B_0 = 0.5$, $D = 0.25$, $h = 2$, and $s_i = 1$ for both nutrients.

13. Initial conditions: all the B_i and N_i at $t = 0$ are uniformly sampled from $[0.05,\ 0.2]$.

With this parameterization, the model equations Eq. (1) can be integrated with any reputable algorithm for solving systems of ordinary differential equations. We used the `NDSolve` routine implemented in Mathematica (Wolfram Research Inc. 2014).

For each of our $10,000$ random parameterizations, the equations were solved until they reached a fixed point. At that point, extinct species were removed, and the Jacobian of the nonextinct part of the system was evaluated, yielding the community matrix.

2 Further results on our simulated matrices

The first thing to note is the overall insensitivity of the binned community matrices to the misclassification rate (i.e., the probability per entry of the community matrix that it gets classified into an incorrect bin). In the main text we always show results with 10% misclassification. Below, we show the same results with 0% and 20% misclassification rates. For the leading eigenvalues (related to stability), we have Figs. S1 and S2 for randomly generated matrices, and Figs. S6 and S7 for Allometric Trophic Networks (Berlow et al. 2009). For the leading eigenvalues of the Hermitian parts (related to reactivity), we have Figs. S3-S5 (randomly generated matrices) and Figs. S8-S10 (Allometric Trophic Networks); here the results with 10% misclassification rate are also included, as these were referred to but not shown in the main text. As can be seen, the sensitivity to an increased misclassification rate is small, though in general it does lead to lower overall prediction accuracy.

We also looked at the effects of connectance on prediction accuracy, both for stability and reactivity (Figs. S11-S18); the effects, however, do not appear systematic. For instance, for binning constant $b = 4$ and number of bins $k = 7$, increasing connectance improves accuracy when the number of species is $S = 50$, but leads to worse accuracy for $S = 500$. This trend is apparent both in the case of stability (Figs. S11, S14) and of reactivity (Figs. S15, S18).

For matrices whose leading eigenvalues are close to the imaginary axis relative to the total spread of the eigenvalues, stability may be incorrectly predicted even if the leading eigenvalue of the binned matrix is not very different from that of the original one. To see how often this would happen, we first take all those matrices for which $|r_A|/\sigma_A < 0.05$, i.e., the magnitude of the leading eigenvalue is less than one twentieth of the total spread of the real parts of all the eigenvalues. The result depends heavily on the binning resolution b, and especially on the number of bins k (Fig. S19). For $k = 3$ (3 bins), stability is more likely to be misclassified than correctly predicted. However, for $b \geq 4$ and $k \geq 5$, stability is correctly predicted in the majority of cases (for instance, exactly 90% of the time when $b = 4$, $k = 7$). If we are more inclusive and consider all those matrices for which $|r_A|/\sigma_A < 0.1$, this number goes up to 97% (Fig. S20).

The same can be done with reactivity (Figs. S21, S22), where we look at the eigenvalues of the Hermitian parts $H(A)$, $H(B)$ of the original and binned matrices A and B. For $k = 7$ bins and binning resolution $b = 4$, reactivity is correctly predicted 92% of the time for matrices with $|r_{H(A)}|/\sigma_{H(A)} < 0.05$, and 99.4% of the time for matrices with $|r_{H(A)}|/\sigma_{H(A)} < 0.1$.

Finally, we show that our results do not depend strongly on the particular form of the probability distribution from which the entries of the random matrices are drawn. We get the same qualitative results if we draw the entries from a lognormal distribution as if we draw them from a Gamma distribution, both for stability (Figs. S23, S24) and for reactivity (Figs. S25, S26).

3 Random matrices

Here we describe in general how random matrices are binned when their entries are drawn independently from the same underlying probability distribution $p_A(x)$. Imagine we are given a binning scheme with k bins whose values are $(x_1, x_2, \ldots, x_k)$. The binned distribution $p_B(x)$ is then given by

$$p_B(x) = \sum_{i=1}^{k} w_i \delta(x - x_i), \tag{5}$$

where $\delta(x - x_i)$ is the Dirac delta function (which can be though of as a normal distribution with mean x_i and zero variance), and w_i is the probability that a given matrix entry will get classified into the ith bin; $\sum_{i=1}^{k} w_i = 1$. This distribution is well-normalized, and its mth moment μ_m has a very simple form:

$$\mu_m = \int_{-\infty}^{+\infty} x^m p_B(x)\,\mathrm{d}x = \sum_{i=1}^{k} w_i \int_{-\infty}^{+\infty} x^m \delta(x - x_i)\,\mathrm{d}x = \sum_{i=1}^{k} w_i x_i^m, \tag{6}$$

where we used the property of the Dirac delta function that $\int_{-\infty}^{+\infty} f(x)\delta(x - x_0)\,\mathrm{d}x = f(x_0)$ for any function $f(x)$. The w_i are calculated from $p_A(x)$ using the criterion that each entry should be lumped into the bin with the closest value. Let us define corresponding integration limits $\Omega_0, \Omega_1, \ldots, \Omega_k$, with $\Omega_0 = -\infty$, $\Omega_k = +\infty$, and $\Omega_i = (x_i + x_{i+1})/2$ for $1 \leq i < k$. We then have

$$w_i = \int_{\Omega_{i-1}}^{\Omega_i} p_A(x)\,\mathrm{d}x. \tag{7}$$

After the binning has been obtained, the eigenvalue distribution can be calculated based on the circular law and its extensions (Ginibre 1965, Girko 1984, Tao et al. 2010, Allesina and Tang 2012). If one is interested in the leading eigenvalue, it can be obtained (Tang et al. 2014) from the formula

$$r = \max\left\{\sqrt{SV} - E + d,\ (S-1)E + d\right\}, \tag{8}$$

where S is the number of rows/columns of the matrix, E is the expected value of $p_A(x)$, V is its variance, and d is a possible constant that has been added to the diagonal of the matrix.

The example given in the main text involves the uniform distribution $p_A(x) = \mathcal{U}[-1,1]$, having mean $E_A = 0$, variance $V_A = 1/3$, and $d = 0$. The leading eigenvalue r_A is therefore simply given by $\sqrt{SV_A}$, or $\sqrt{S/3}$. We bin this matrix with three bins $(-1,\ 0,\ 1)$. The probability of a given $x \in \mathcal{U}[-1,1]$ falling into the -1 or 1 bins is $1/4$, while the probability of falling into the 0 bin is $1/2$. We therefore have $w_1 = 1/4$, $w_2 = 1/2$, and $w_3 = 1/4$. The binned distribution $p_B(x)$ therefore reads

$$p_B(x) = \sum_{i=1}^{3} w_i \delta(x - x_i) = \frac{\delta(x+1) + 2\delta(x) + \delta(x-1)}{4}. \tag{9}$$

The mean of $p_B(x)$ is

$$E_B = \int_{-\infty}^{+\infty} x\, p_B(x)\, \mathrm{d}x = \frac{1}{4}\int_{-\infty}^{+\infty} x\left[\delta(x+1) + 2\delta(x) + \delta(x-1)\right] \mathrm{d}x = \frac{1}{4}\left(1 + 2\cdot 0 + (-1)\right) = 0, \tag{10}$$

while its variance is

$$\begin{aligned} V_B &= \int_{-\infty}^{+\infty} (x - E_B)^2\, p_B(x)\, \mathrm{d}x = \int_{-\infty}^{+\infty} x^2 p_B(x)\, \mathrm{d}x \\ &= \frac{1}{4}\int_{-\infty}^{+\infty} x^2 \left[\delta(x+1) + 2\delta(x) + \delta(x-1)\right] \mathrm{d}x = \frac{1}{4}\left(1^2 + 2\cdot 0^2 + (-1)^2\right) = \frac{1}{2}. \end{aligned} \tag{11}$$

The leading eigenvalue r_B of the binned matrix is then given by Eq. (8), which in this case simplifies to $r_B = \sqrt{SV_B} = \sqrt{S/2}$. The ratio of r_B and r_A is then

$$\frac{r_B}{r_A} = \frac{\sqrt{S/2}}{\sqrt{S/3}} = \sqrt{\frac{3}{2}} \approx 1.22. \tag{12}$$

The same calculation and comparison can be made with a different binning scheme involving five bins $(-1,\ -1/2,\ 0,\ 1/2,\ 1)$. Using Eq. (7), we obtain $w_i = (1,\ 2,\ 2,\ 2,\ 1)/8$, therefore $p_B(x)$ is given by

$$p_B(x) = \sum_{i=1}^{5} w_i \delta(x - x_i) = \frac{\delta(x+1) + 2\delta(x+1/2) + 2\delta(x) + 2\delta(x-1/2) + \delta(x-1)}{8}. \tag{13}$$

The mean is again

$$E_B = \int_{-\infty}^{+\infty} x\, p_B(x)\, \mathrm{d}x = 0, \tag{14}$$

but the variance now reads

$$\begin{aligned} V_B &= \int_{-\infty}^{+\infty} (x-E_B)^2\, p_B(x)\,\mathrm{d}x = \int_{-\infty}^{+\infty} x^2 p_B(x)\,\mathrm{d}x \\ &= \frac{1}{8}\int_{-\infty}^{+\infty} x^2\left[\delta(x+1)+2\delta(x+1/2)+2\delta(x)+2\delta(x-1/2)+\delta(x-1)\right]\mathrm{d}x = \frac{3}{8}. \end{aligned} \tag{15}$$

The new ratio of the leading eigenvalues, based on Eq. (8), is

$$\frac{r_B}{r_A} = \frac{\sqrt{3S/8}}{\sqrt{S/3}} = \sqrt{\frac{9}{8}} \approx 1.06. \tag{16}$$

All the above formulas assumed that the matrix entries were all independently drawn from the same distribution. The same approach may in principle be extended to different types of random matrices. For instance, the elliptic law (Sommers et al. 1998, Allesina and Tang 2012, Tang et al. 2014) concerns the eigenvalues of matrices where all symmetric pairs of entries (A_{ij},A_{ji}) are sampled from some bivariate probability distribution with identically distributed marginals and correlation ρ. The eigenvalues are then uniformly distributed within an ellipse of horizontal and vertical semi-axes $\sqrt{SV}(1+\rho)$ and $\sqrt{SV}(1-\rho)$, respectively. The leading eigenvalue r is then equal to

$$r = \max\left\{(1+\rho)\sqrt{SV} - E + d,\ (S-1)E + d\right\}, \tag{17}$$

where d is the mean of the diagonal entries, $E = \overline{A_{ij}}$ is the expected value of the offdiagonal entries, $V = \mathrm{Var}(A_{ij})$ is their variance, and

$$\rho = \frac{\overline{A_{ij}A_{ji}} - E^2}{V} \tag{18}$$

is the expected correlation between symmetric pairs of (offdiagonal) entries. To determine the ratio of the leading eigenvalues of the original matrix A and the binned matrix B, one simply calculates E_A, V_A, ρ_A and E_B, V_B, ρ_B, applies Eq. (17) to obtain both r_A and r_B, and calculates their ratio. There is one difference in how $p_B(x,y)$ is calculated, however: the integrals in Eq. (7) now become double integrals, going over both directions of the bivariate distribution $p_A(x,y)$.

4 Calculating pseudospectral contour lines

There are several methods for computing the pseudospectra of matrices (Trefethen and Embree 2005, chapters 39-44), each adapted to different situations. Here we only describe the simplest algorithm (Trefethen and Embree 2005, p. 371), which we used to generate the pseudospectral contour plots of the main text.

The ε-pseudospectrum of a matrix A is defined as the set of complex numbers that are eigenvalues of all possible perturbed matrices $A+P$ with $\|P\| < \varepsilon$, where the matrix norm $\|\cdot\|$ is defined as $\|P\| = \sqrt{\lambda_{\max}(P^*P)}$, with P^* being the conjugate transpose of P, and $\lambda_{\max}(P^*P)$ the largest eigenvalue of P^*P. An equivalent definition uses the concept of singular values (the singular values

of a matrix A are the square roots of the eigenvalues of A^*A): the ε-pseudospectrum of A is the set of complex numbers z such that the smallest singular value of the matrix $zI - A$ is smaller than ε, where I is the identity matrix (Trefethen and Embree 2005, pp. 16-17). The simplest way of obtaining pseudospectra is to compute the smallest singular values of $zI - A$ on a regular grid of points in the complex plane and then visualize this information via a contour plot.

If the matrix to be analyzed is given by `A`, the grid consists of `m` points in both the real and imaginary directions, and the starting and endpoints along the real and imaginary directions are given, respectively, by `reMin`, `reMax`, `imMin`, and `imMax`, the following R code (R Development Core Team 2008) computes the minimum singular values on the grid and stores them in the matrix `sigmin`:

```
x <- seq(from=reMin, to=reMax, len=m)
y <- seq(from=imMin, to=imMax, len=m)
sigmin <- matrix(0, m, m)
for (k in 1:m) {
  for (j in 1:m) {
    sigmin[j,k] <- min(svd((x[k]+y[j]*1i)*diag(ncol(A))-A, nu=0, nv=0)$d)
  }
}
```

This information may then be visualized by, say, invoking the `contour` function. Note that there are more efficient algorithms available; see the book by Trefethen and Embree (2005, p. 375) for one that is much faster but still reasonably simple.

5 Departure from normality

Despite the avaliability of algorithms for obtaining pseudospectral contour lines, the computation itself can be quite time-intensive, especially when S is large and the resolution of the grid on the complex plane is high. It would be good to have a way of assessing, before performing the computations, whether one expects large or small pseudospectral regions. Since normal matrices possess the smallest possible pseudospectra (Trefethen and Embree 2005, Theorem 2.2), a natural way to see if the spectrum of a matrix would be sensitive to perturbations is to measure the matrix's departure from normality. Any such metric is bound to be approximate, because nonnormality is a complex property that cannot be encompassed into a single number (Trefethen and Embree 2005, p. 446). Nevertheless, such measures do provide a quick and useful way of checking whether we expect a matrix to be oversensitive to perturbations.

One commonly used measure of the departure from normality (Henrici 1962, Lee 1996) is

$$\mathrm{dep}(A) = \sqrt{\sum_{i=1}^{S} \sigma_i^2 - \sum_{i=1}^{S} |\lambda_i|^2}, \tag{19}$$

where λ_i and σ_i are the ith eigenvalue and singular value of A, respectively (singular values are the square roots of the eigenvalues of A^*A). This metric is equal to zero for normal matrices, and can get arbitrarily large for highly nonnormal ones. As there is no upper limit to $\mathrm{dep}(A)$, it is not immediately clear whether any given nonzero value of this metric should be considered large or small.

One way of finding a point of comparison is to calculate $\mathrm{dep}(M)$ for a simple $S \times S$ random matrix M, where each entry is drawn independently from the same distribution with mean zero and variance V. The reason is that such matrices are known to possess only a very mild degree of nonnormality, being fairly robust to perturbations of their entries (Edelman 1988). Any matrix whose departure from normality is comparable to or smaller than that of the aforementioned random matrix is therefore likely to possess a spectrum that does not change much in response to binning. Note that the eigenvalue and singular value distributions of these random matrices do not depend on the shape of the probability distribution from which their entries are drawn, merely its variance.

The departure from normality of the random matrix described above can be derived analytically. The probability distribution $p_\sigma(x)$ of their singular values follows

$$p_\sigma(x) = \frac{2}{\pi\sqrt{SV}}\sqrt{1-\frac{x^2}{4SV}} \tag{20}$$

for $0 \leq x \leq 2\sqrt{SV}$ and 0 otherwise (Marčenko and Pastur 1967). In turn, the probability distribution of the absolute values of their eigenvalues $p_\lambda(x)$, as a function of the distance x from the origin of the complex plane, follows

$$p_\lambda(x) = \frac{2x}{SV} \tag{21}$$

for $0 \leq x \leq \sqrt{SV}$ and 0 otherwise (Tao et al. 2010, Allesina and Tang 2012).

Eq. (19) instructs us to sum the squared singular values and eigenvalues. We therefore transform $p_\sigma(x)$ and $p_\lambda(x)$ by introducing the new variable $y = x^2$, from which $x = \sqrt{y}$ and $|\mathrm{d}x/\mathrm{d}y| = \left(2\sqrt{y}\right)^{-1}$. The transformed probability distribution $g_\sigma(y)$ of singular values then reads

$$g_\sigma(y) = p_\sigma\left(x(y)\right)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = \frac{1}{\pi\sqrt{SVy}}\sqrt{1-\frac{y}{4SV}} \tag{22}$$

for $0 \leq y \leq 4SV$ and 0 otherwise. In turn, the transformed probability distribution $g_\lambda(y)$ of the absolute eigenvalues reads

$$g_\lambda(y) = p_\lambda\left(x(y)\right)\left|\frac{\mathrm{d}x}{\mathrm{d}y}\right| = \frac{1}{SV} \tag{23}$$

for $0 \leq y \leq SV$ and 0 otherwise.

We can now evaluate Eq. (19) by approximating the sums with integrals (this assumes S is large). The sum of all the squared singluar values is equal to S times their mean:

$$\sum_{i=1}^{S}\sigma_i^2 = S\int_0^{4SV} y\,g_\sigma(y)\,\mathrm{d}y = S\int_0^{4SV}\frac{y}{\pi\sqrt{SVy}}\sqrt{1-\frac{y}{4SV}} = S^2V. \tag{24}$$

The sum of the absolute squares of all eigenvalues is again S times their mean:

$$\sum_{i=1}^{S} |\lambda_i|^2 = S \int_0^{SV} y\, g_\lambda(y)\, \mathrm{d}y = S \int_0^{SV} \frac{y}{SV}\, \mathrm{d}y = \frac{S^2 V}{2}. \tag{25}$$

For a random matrix M, Eq. (19) then reads

$$\mathrm{dep}(M) = \sqrt{\sum_{i=1}^{S} \sigma_i^2 - \sum_{i=1}^{S} |\lambda_i|^2} = \sqrt{S^2 V - \frac{S^2 V}{2}} = S\sqrt{\frac{V}{2}}. \tag{26}$$

We can now introduce a measure of the *scaled departure from normality* $\mathrm{depn}(A)$, comparing the degree of nonnormality with that of a random matrix which is of the same size and its entries are of the same variance as the original matrix. Take an $S \times S$ matrix A and determine the sample variance V of all its entries. The metric then reads

$$\mathrm{depn}(A) = \frac{\mathrm{dep}(A)}{S\sqrt{V/2}}. \tag{27}$$

Whenever this metric is equal to one, the degree of nonnormality of A is that of a random matrix.

The following R function (R Development Core Team 2008) calculates this metric, given that the sample variance of the entries of A is nonzero:

```
depn <- function(A) {
  eA <- eigen(A, only.values=TRUE)$values
  sA <- svd(A, nu=0, nv=0)$d
  depA <- sqrt(sum(sA^2) - sum(abs(eA)^2))
  S <- nrow(A)
  V <- var(as.vector(A))
  depRand <- S*sqrt(V/2)
  return(depA/depRand)
}
```

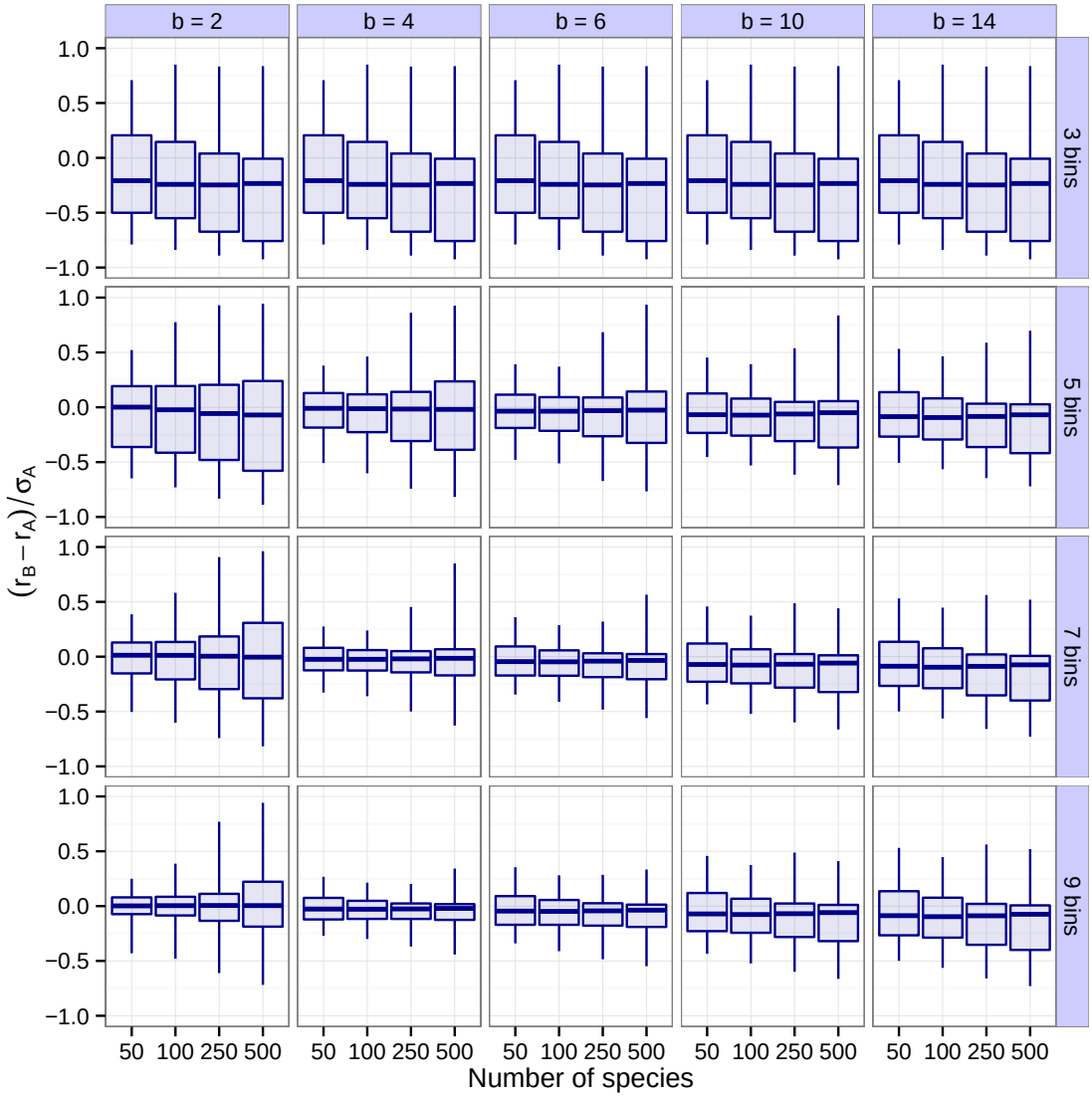

Figure S1: Box plots of how the leading eigenvalues of randomly generated community matrices A are captured by those of their binned counterparts B. Each matrix is binned with no misclassification. Rows correspond to different values of the binning resolution b; columns to different numbers of bins k. The data in each panel are separated based on the number of species. The ordinate of the panels shows the difference between the leading eigenvalue r_A of the original and r_B of the binned matrices relative to σ_A, the total range of the real parts of the original matrix's eigenvalues. Interpretation of the box plots: median (lines), 5% to 95% quantiles (boxes; note that they encompass 90% instead of the usual 50% of the data), and ranges (whiskers).

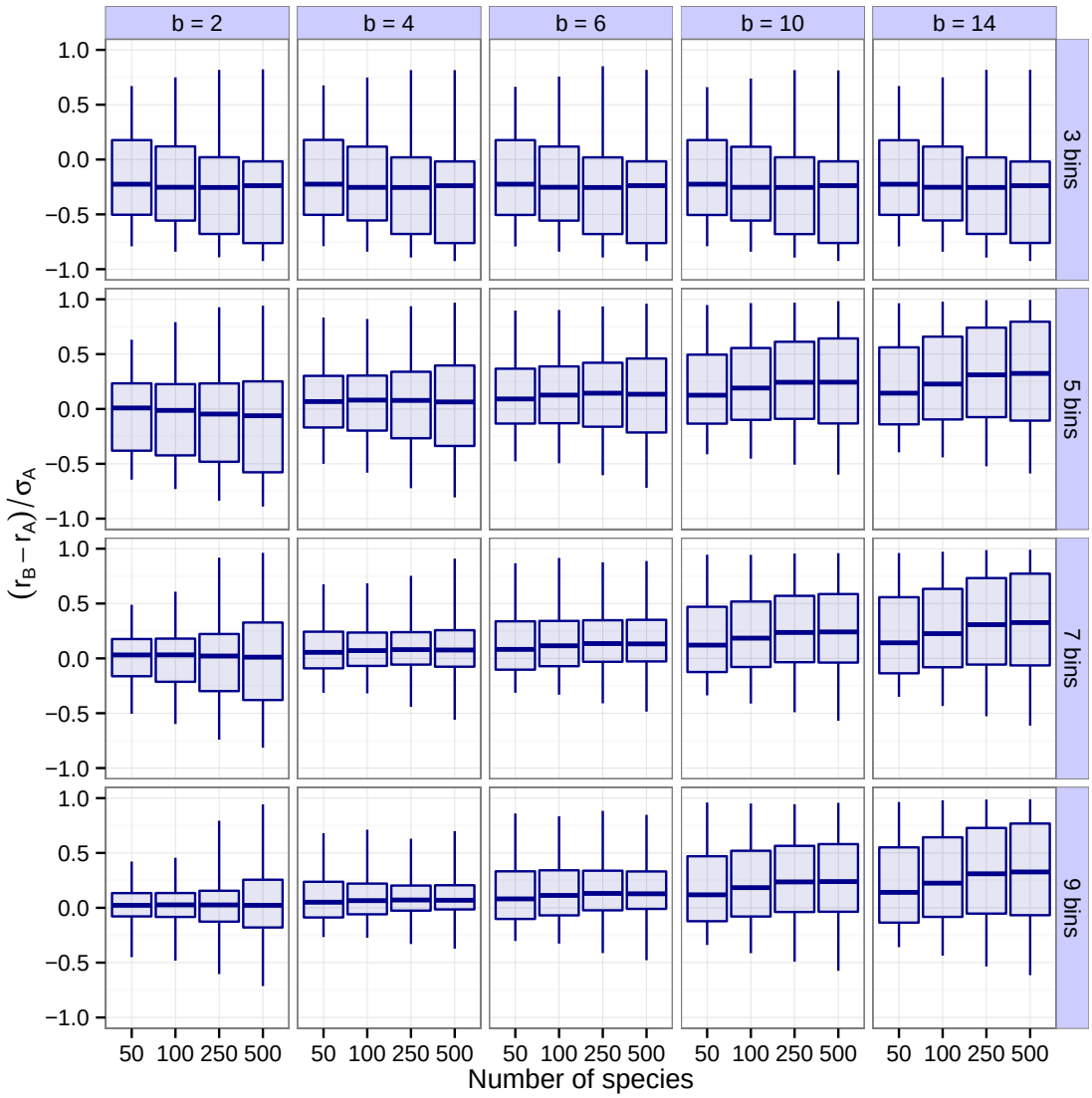

Figure S2: Box plots of how the leading eigenvalues of randomly generated community matrices A are captured by those of their binned counterparts B. Each matrix is binned with a 20% misclassification rate. Otherwise, the plot is organized just like Fig. S1.

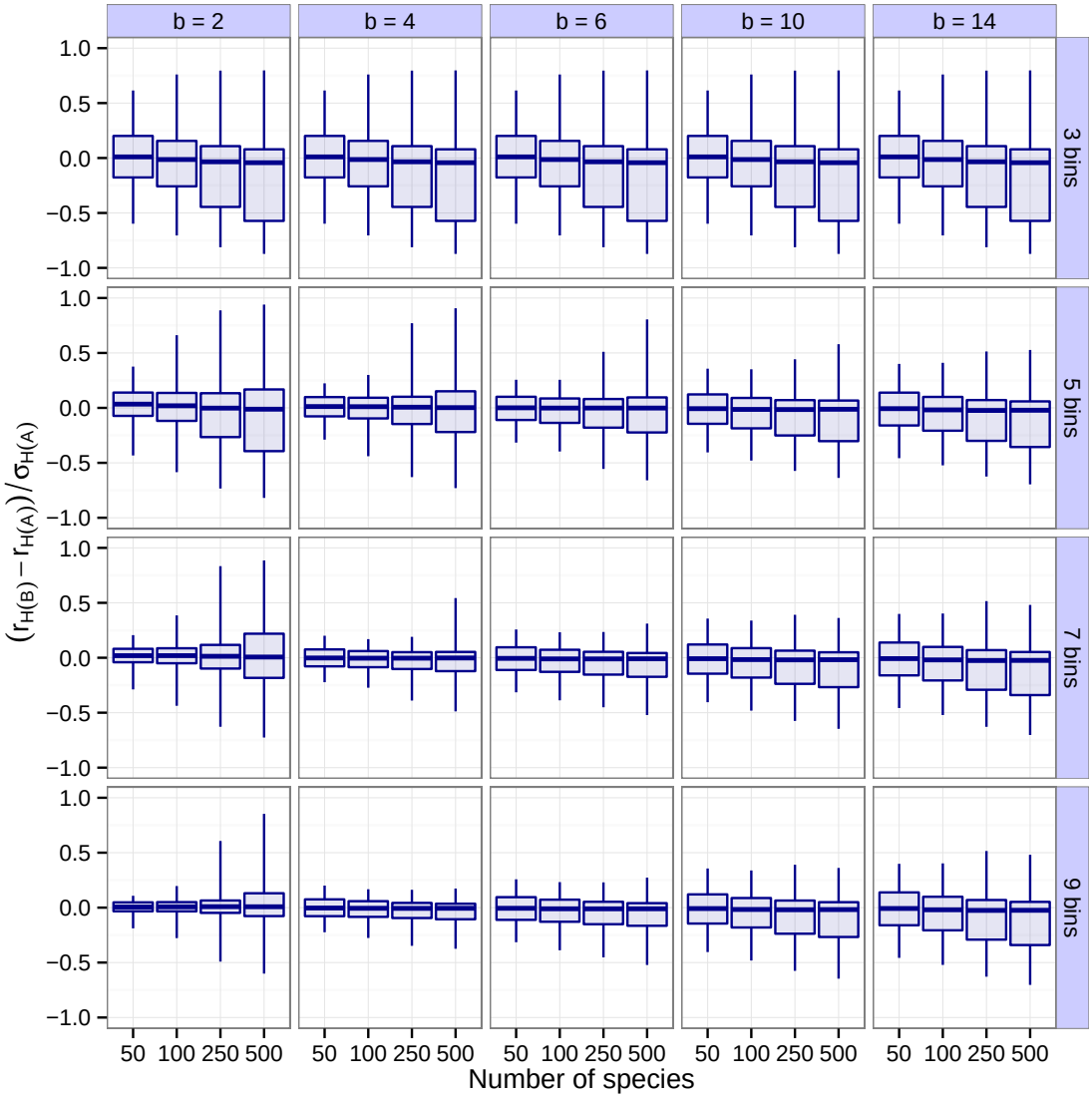

Figure S3: Box plots of how well the leading eigenvalue of A's Hermitian part $H(A)$ is approximated by the leading eigenvalue of B's Hermitian part $H(B)$, where B is the binned counterpart of A. The ordinate of each panel shows the difference between the leading eigenvalues $r_{H(A)}$ of the Hermitian parts of the original and $r_{H(B)}$ of the Hermitian parts of the binned matrices relative to $\sigma_{H(A)}$, the total range of the eigenvalues of the Hermitian parts of the original matrices. The rate of misclassification is 0%. Rows, columns, and panels are organized just as in Fig. S1.

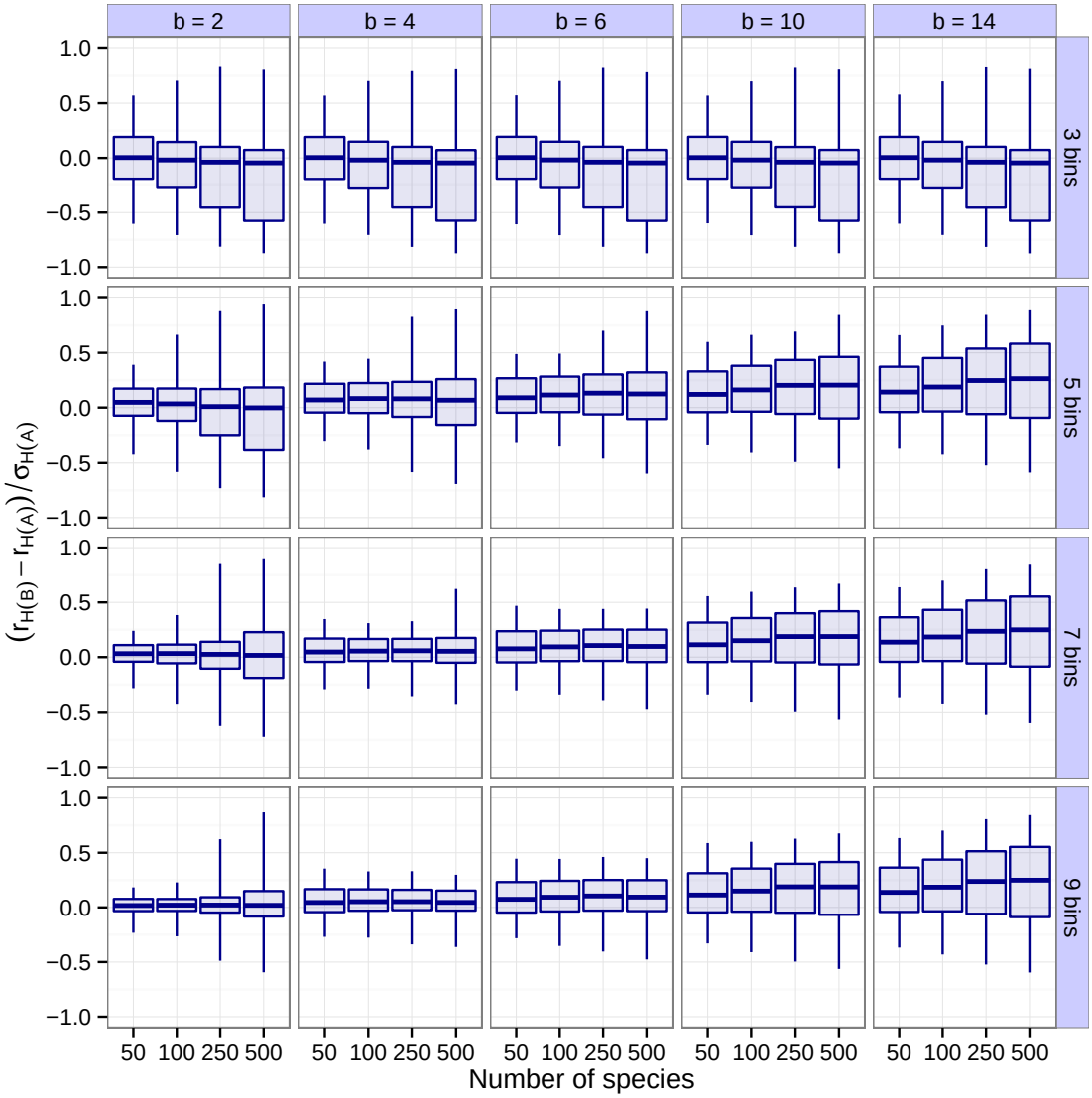

Figure S4: As Fig. S3, except each matrix is binned with a 10% misclassification rate.

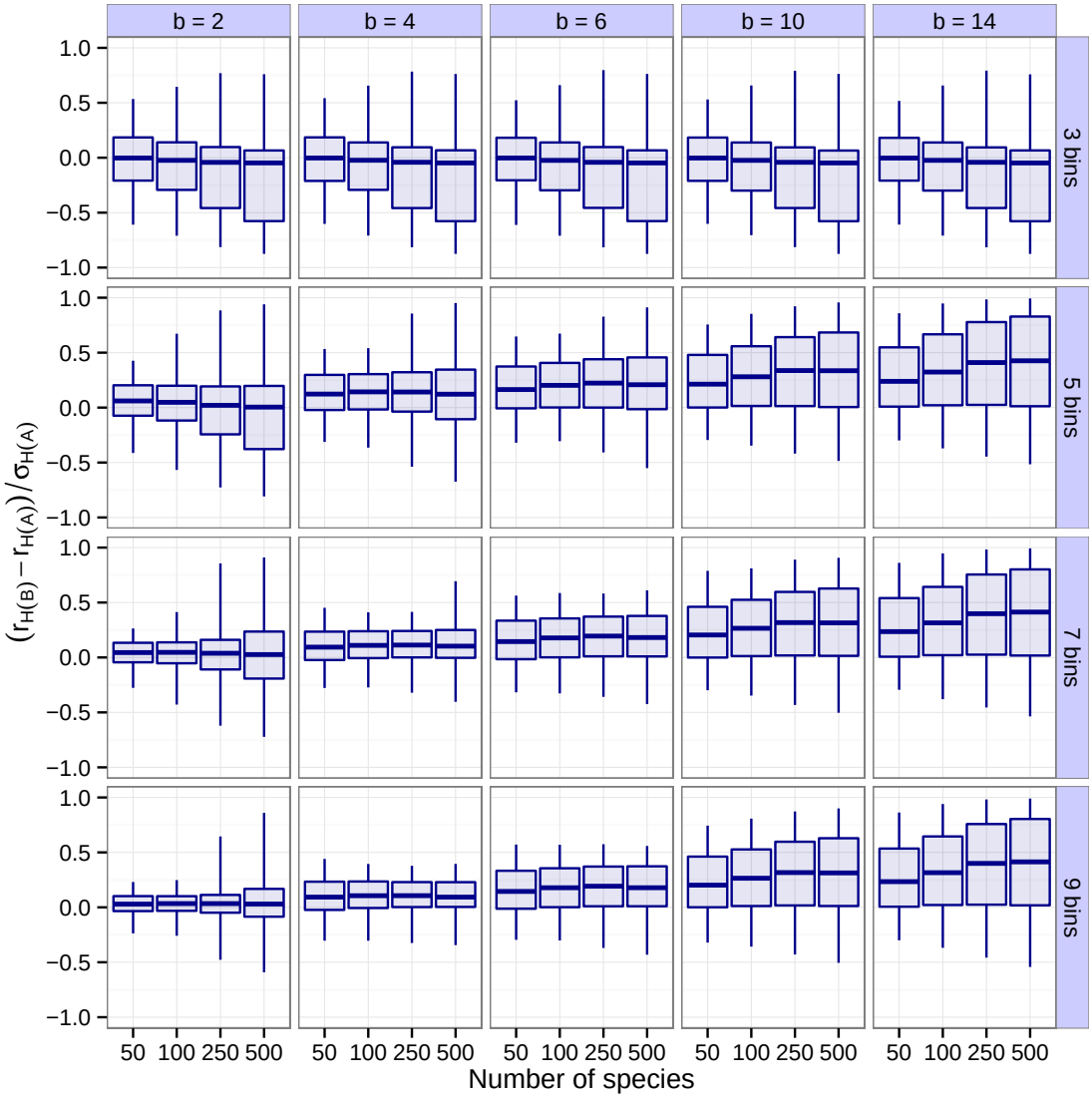

Figure S5: As Fig. S3, except each matrix is binned with a 20% misclassification rate.

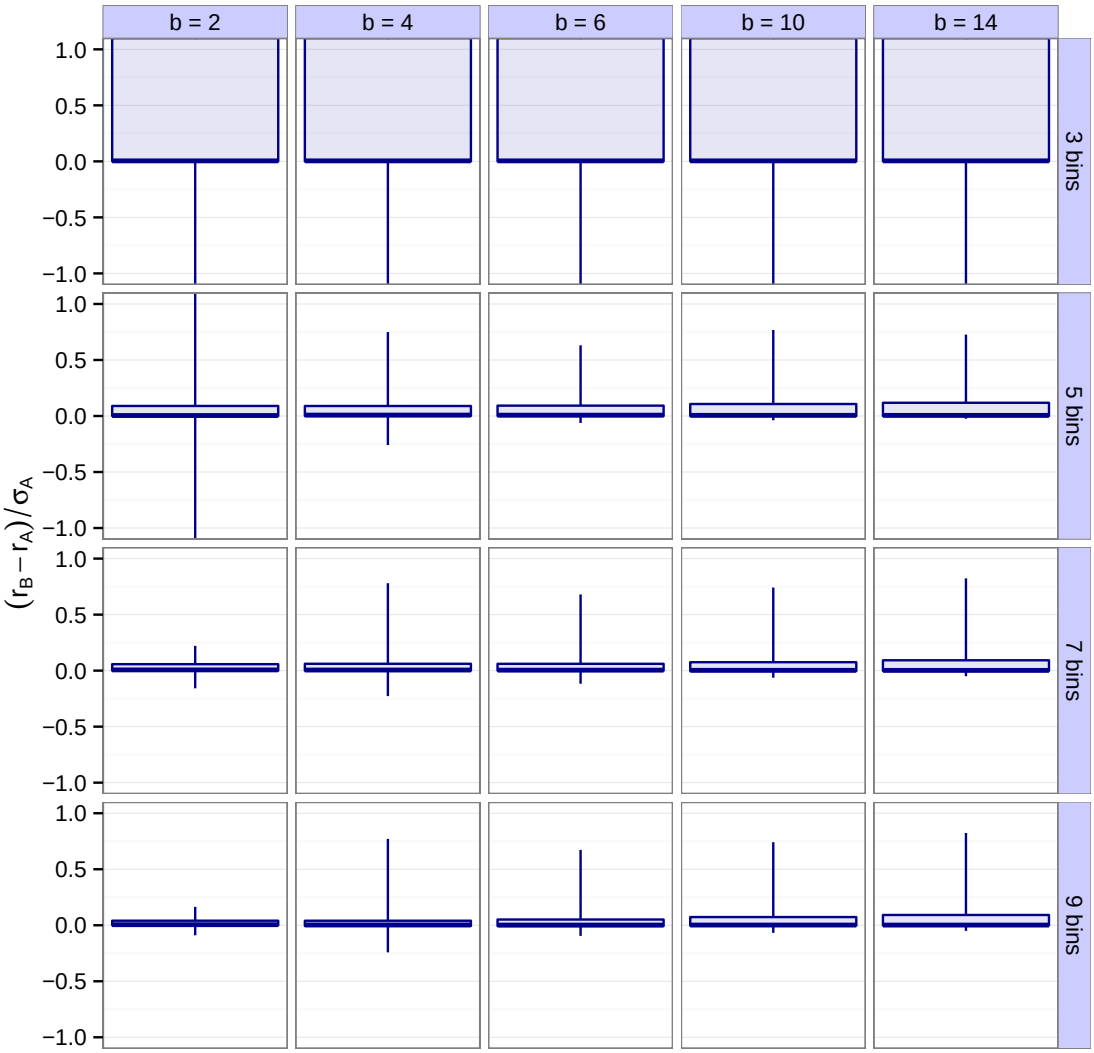

Figure S6: Box plots of how the leading eigenvalues of community matrices A are captured by those of their binned counterparts B, where the matrices A are generated by the Allometric Trophic Network model (Berlow et al. 2009). Each matrix is binned with no misclassification. The figure is organized just like Fig. S1, except the data in the panels are not separated based on the number of species.

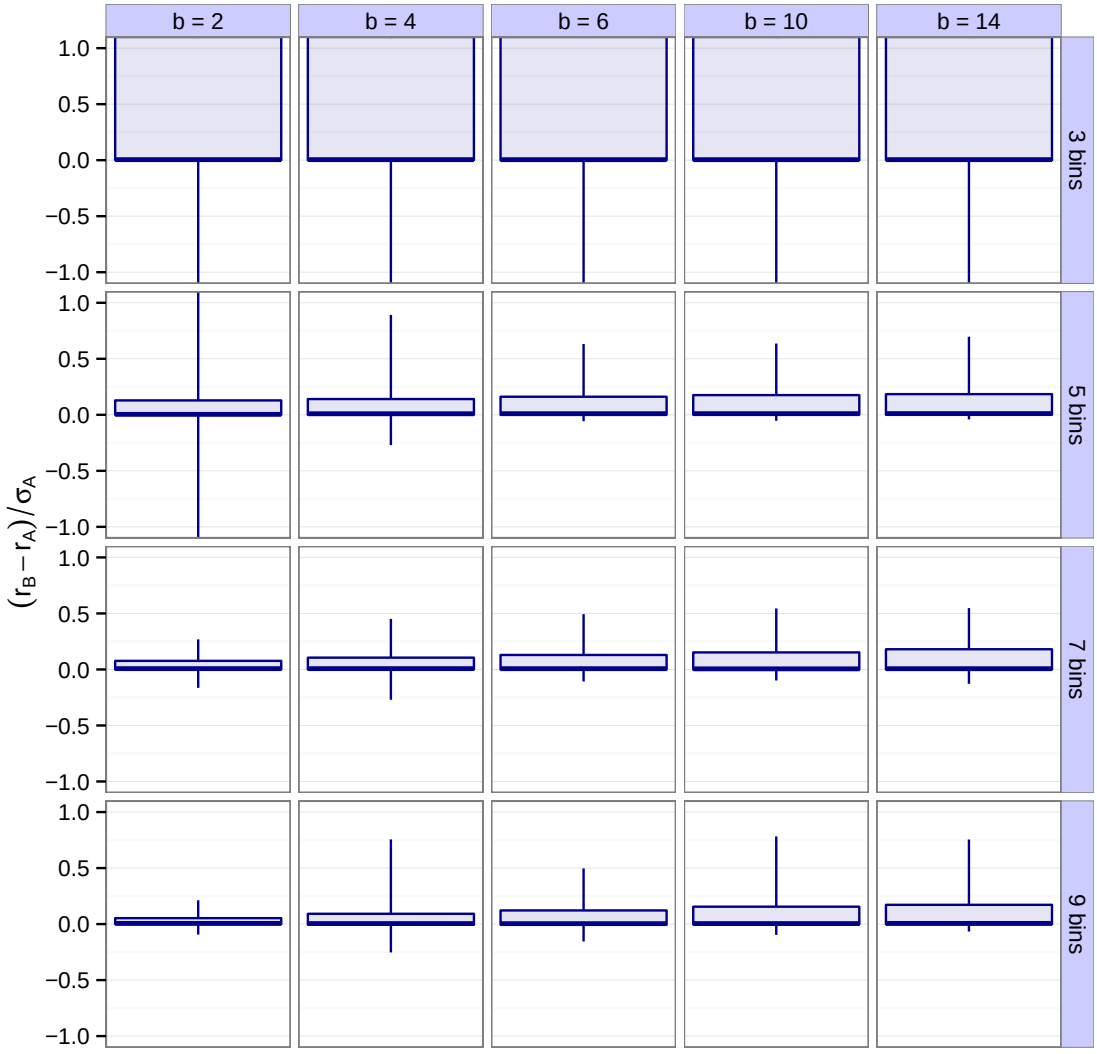

Figure S7: As Fig. S6, but with a 20% misclassification rate.

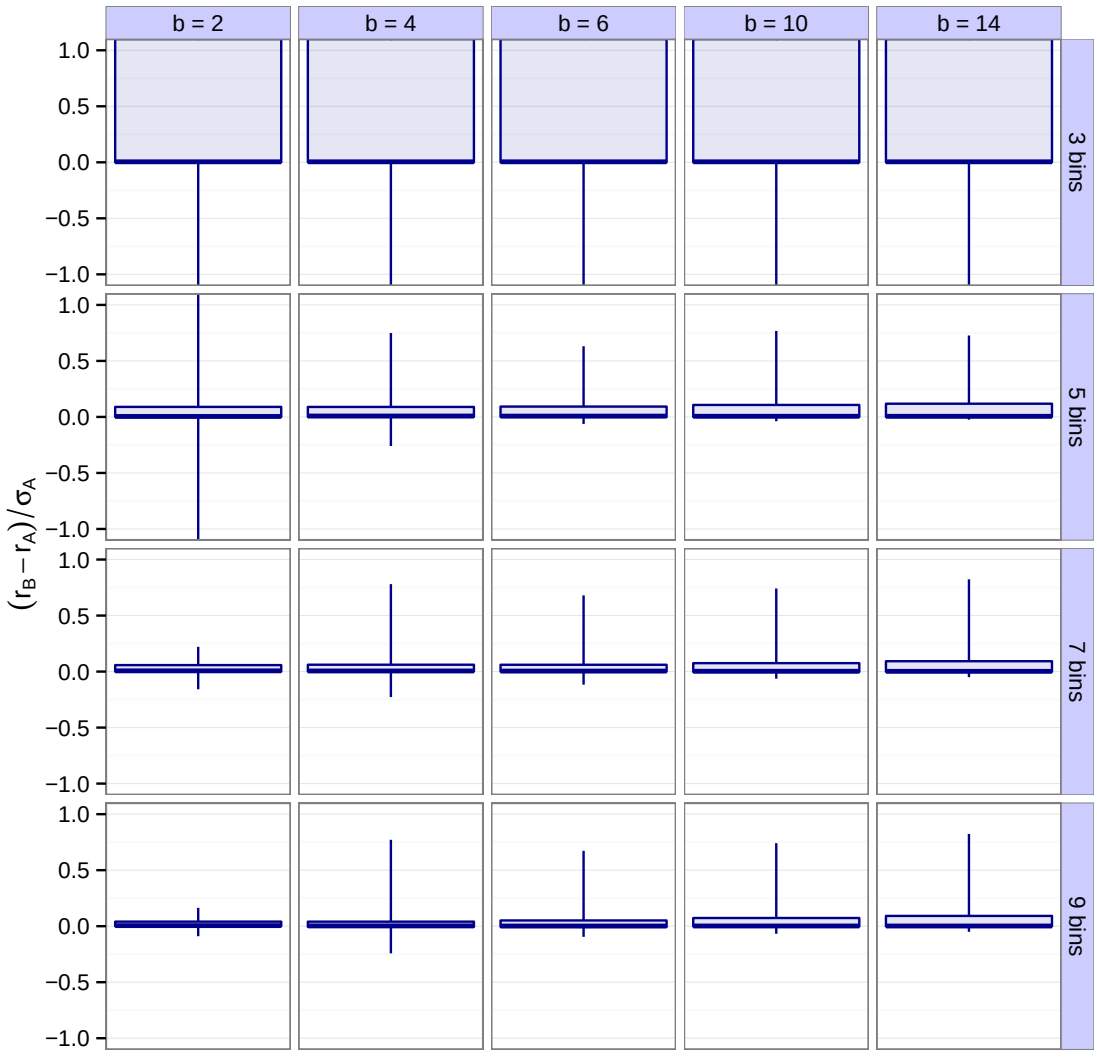

Figure S8: As Fig. S6, but for reactivity instead of stability; i.e., how well the leading eigenvalue of A's Hermitian part $H(A)$ is approximated by the leading eigenvalue of B's Hermitian part $H(B)$ in the Allometric Trophic Network model. Each matrix is binned with zero rate of misclassification.

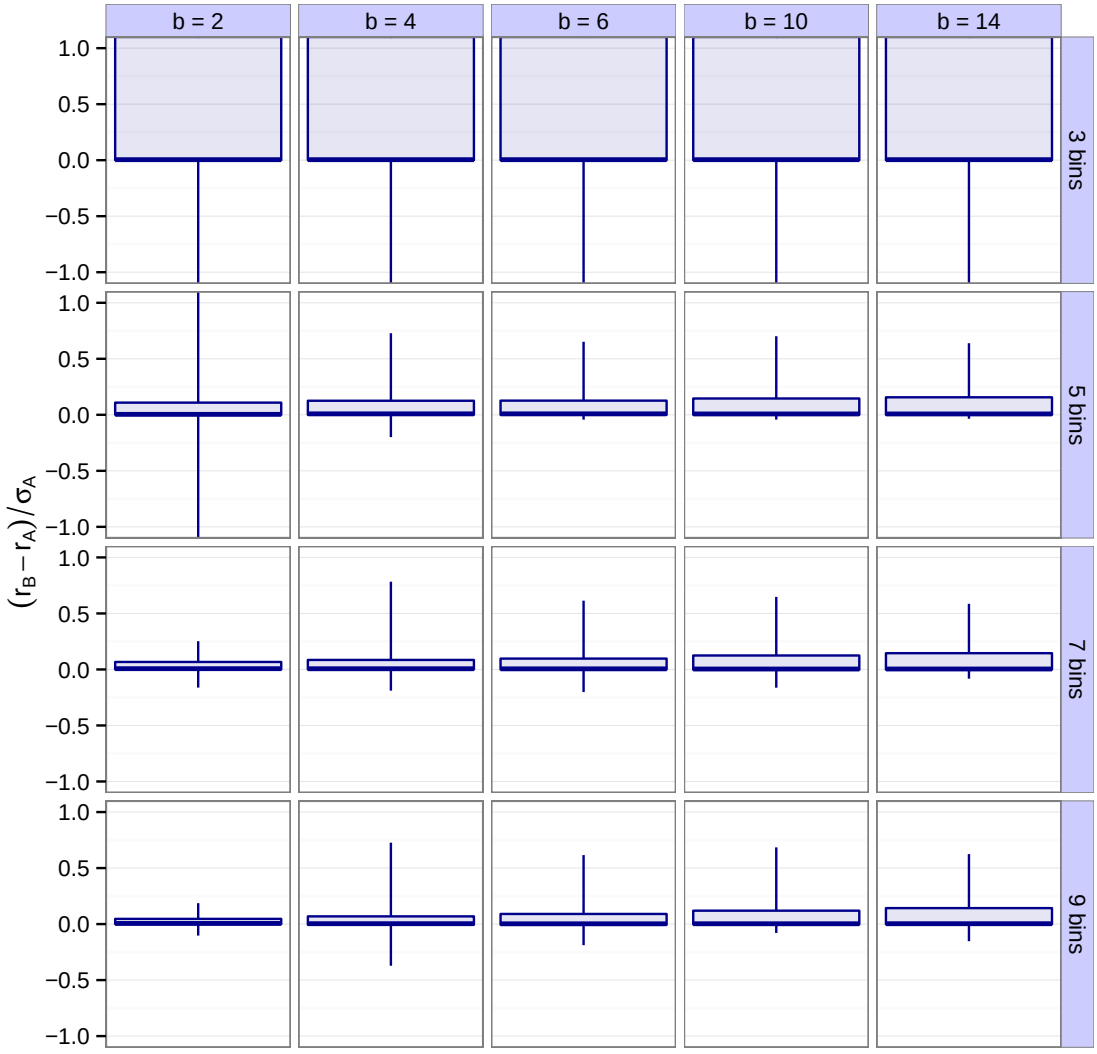

Figure S9: As Fig. S8, but with a 10% misclassification rate.

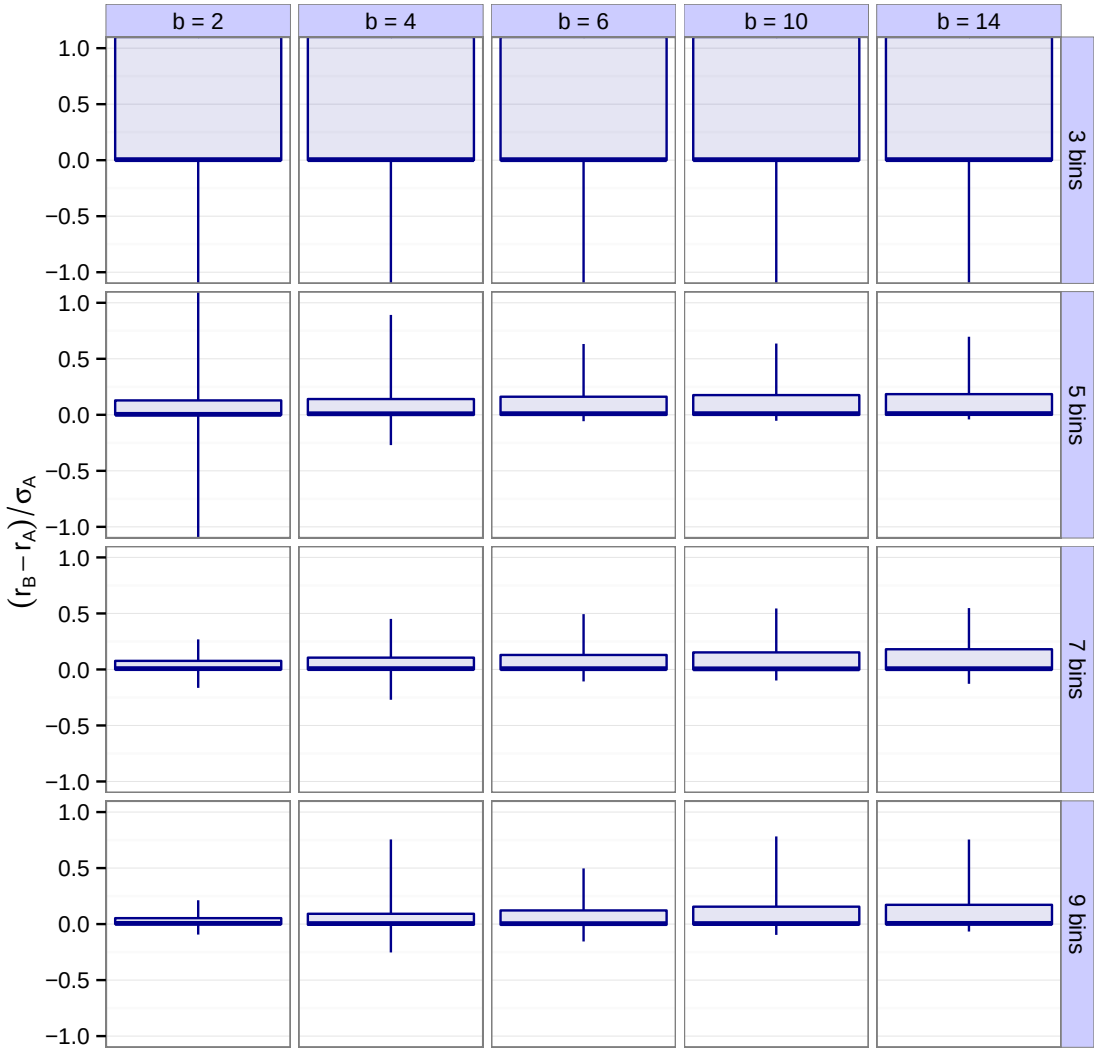

Figure S10: As Fig. S8, but with a 20% misclassification rate.

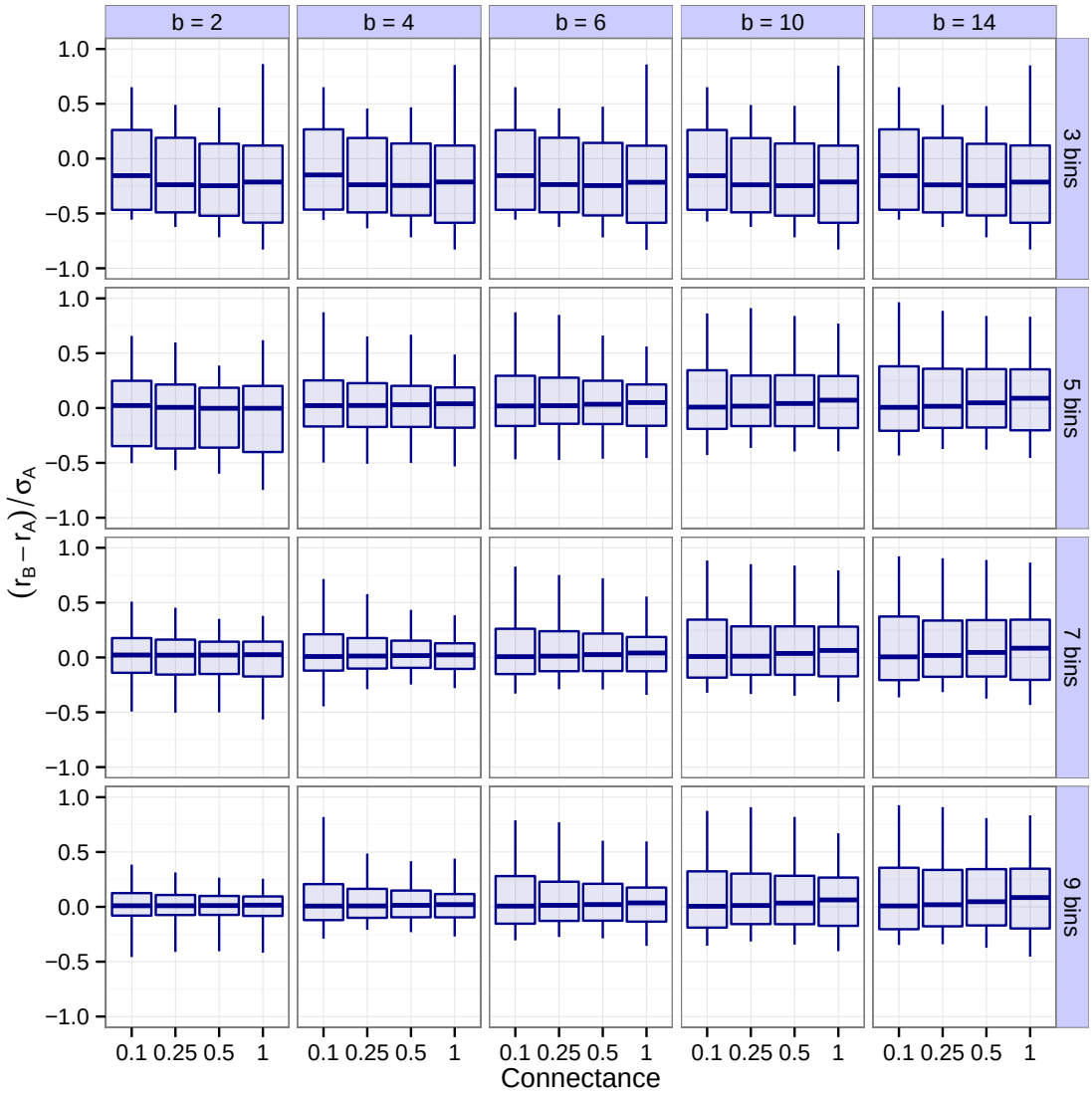

Figure S11: The effect of varying network connectance on the prediction accuracy of the leading eigenvalue. Organized as Fig. S1, except with 10% misclassification rate, the abscissa is organized by different values of the connectance instead of different values of the number of species, and the number of species is fixed at $S = 50$.

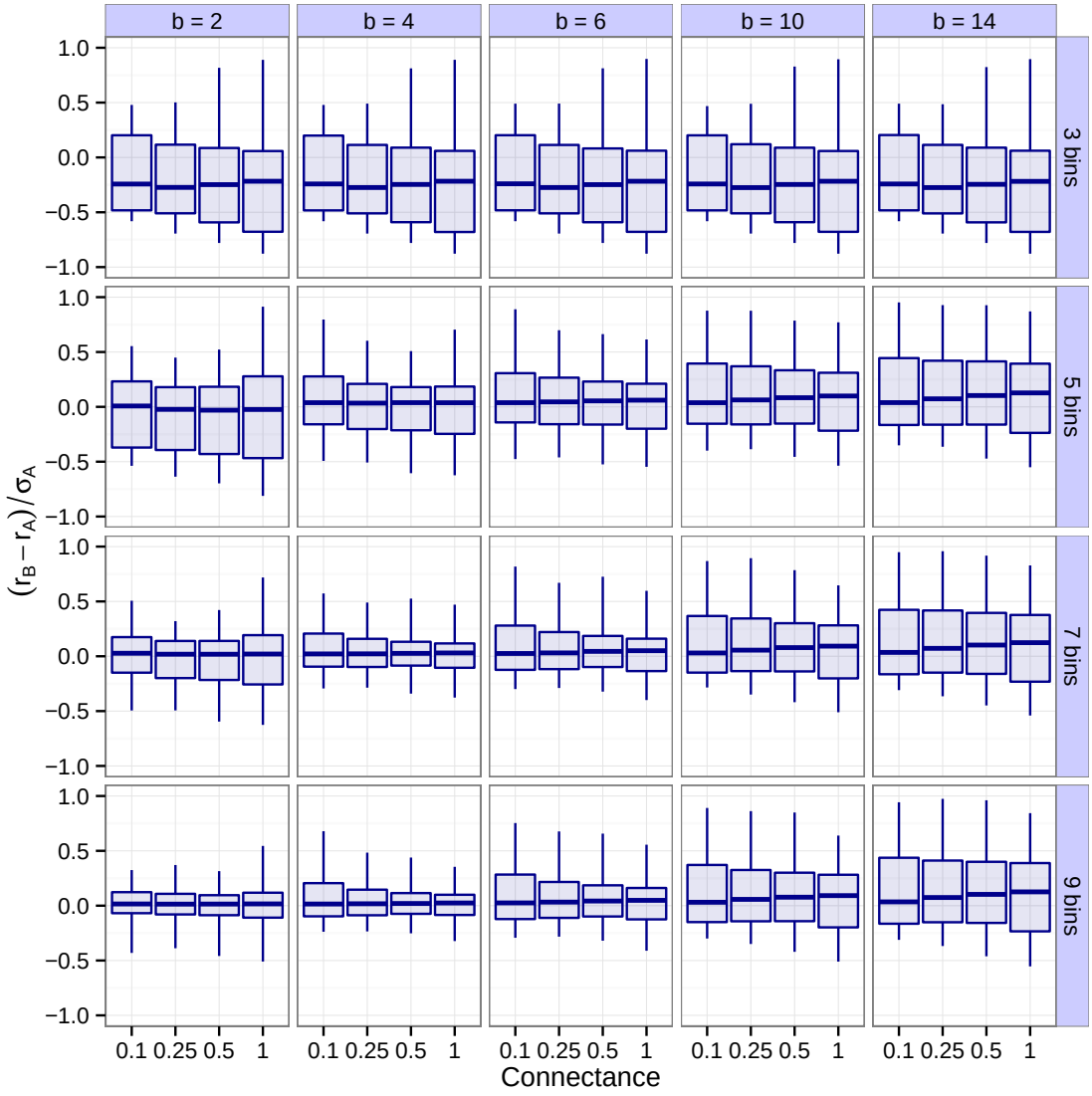

Figure S12: As Fig. S11, but with the number of species fixed at $S = 100$.

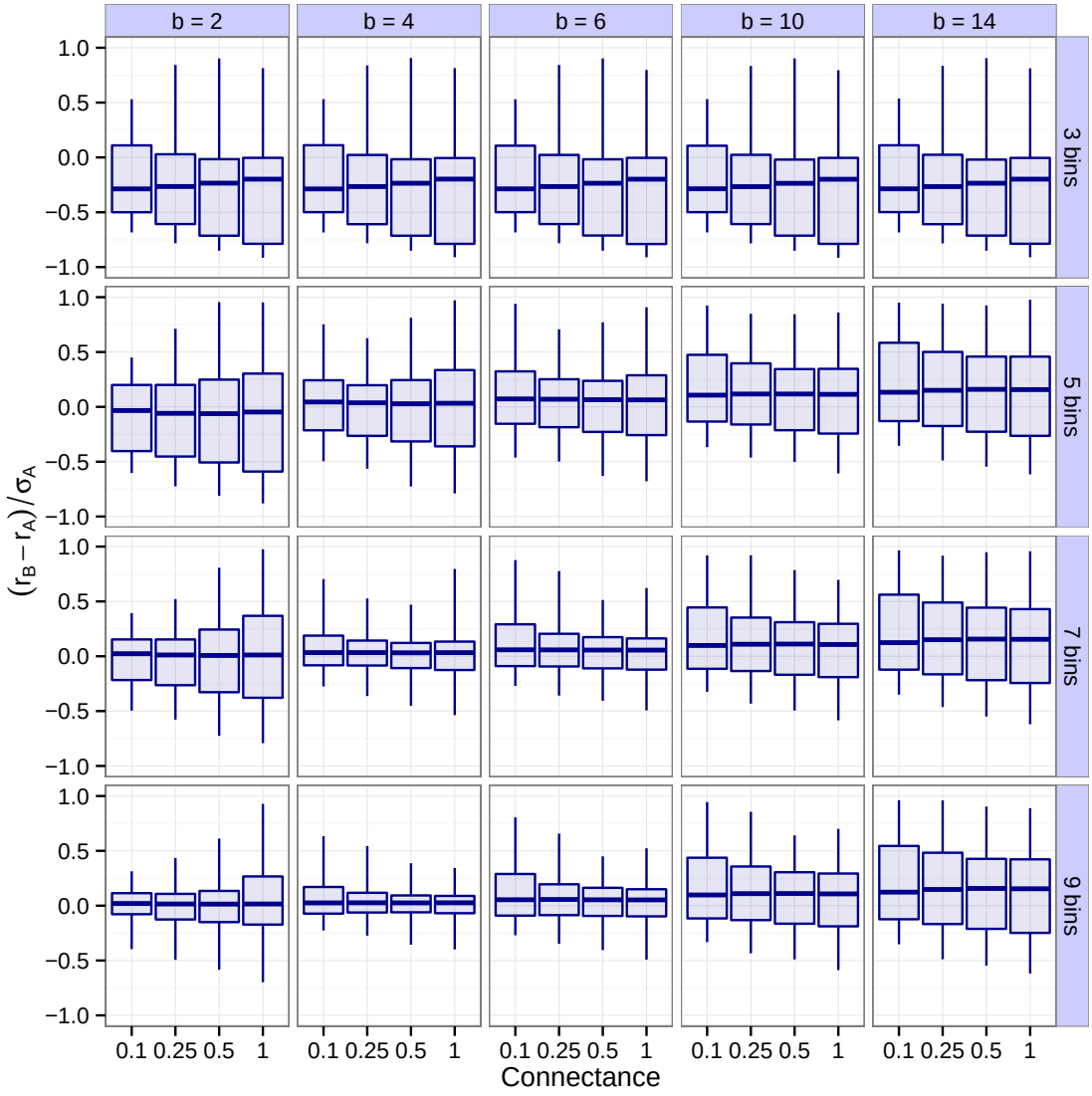

Figure S13: As Fig. S11, but with the number of species fixed at $S = 250$.

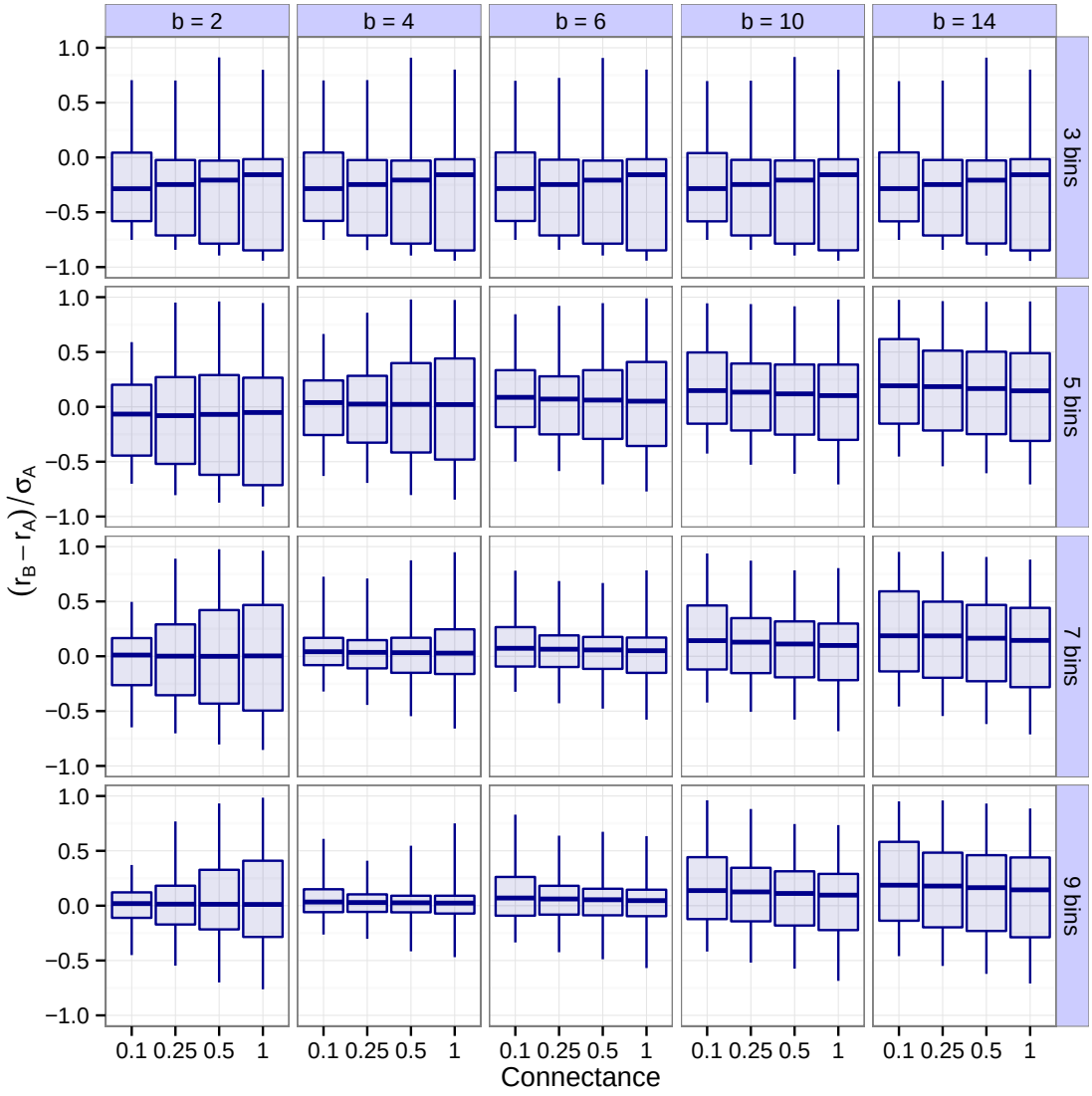

Figure S14: As Fig. S11, but with the number of species fixed at $S = 500$.

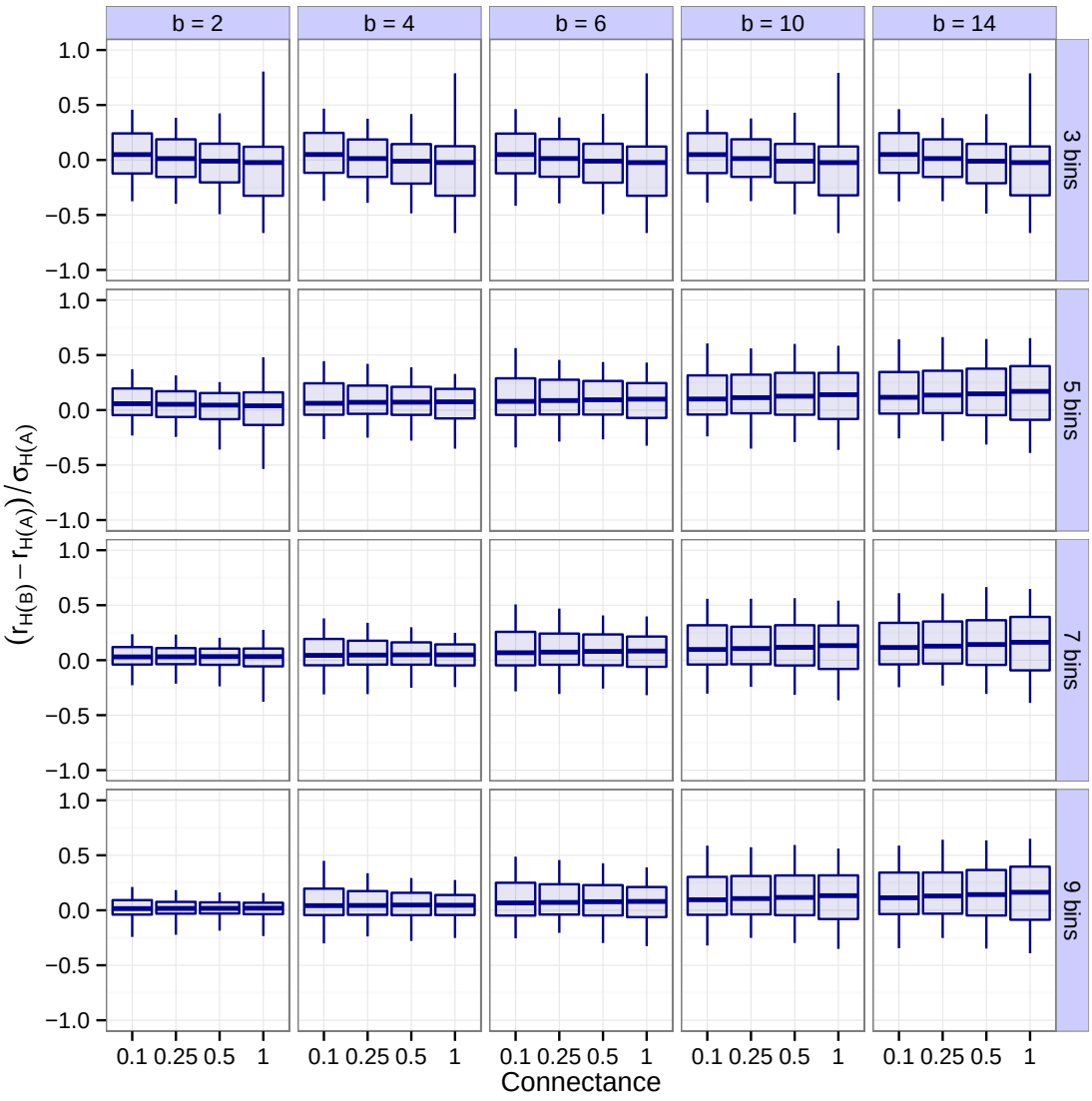

Figure S15: The effect of varying network connectance on the prediction accuracy of the leading eigenvalue of the Hermitian part (related to reactivity). As Fig. S11, but the ordinate represents predictions of reactivity instead of stability.

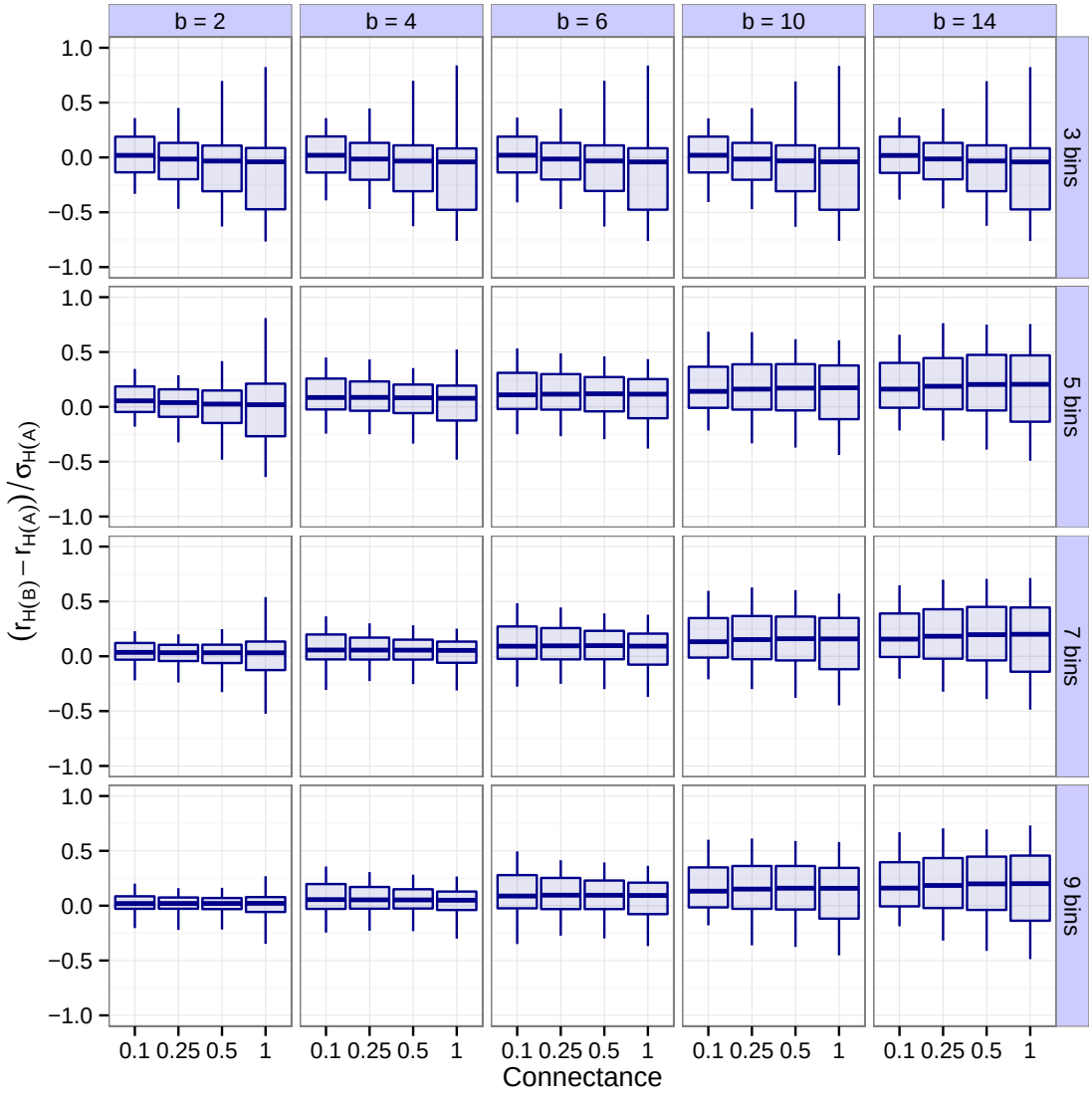

Figure S16: As Fig. S11, but with the number of species fixed at $S = 100$.

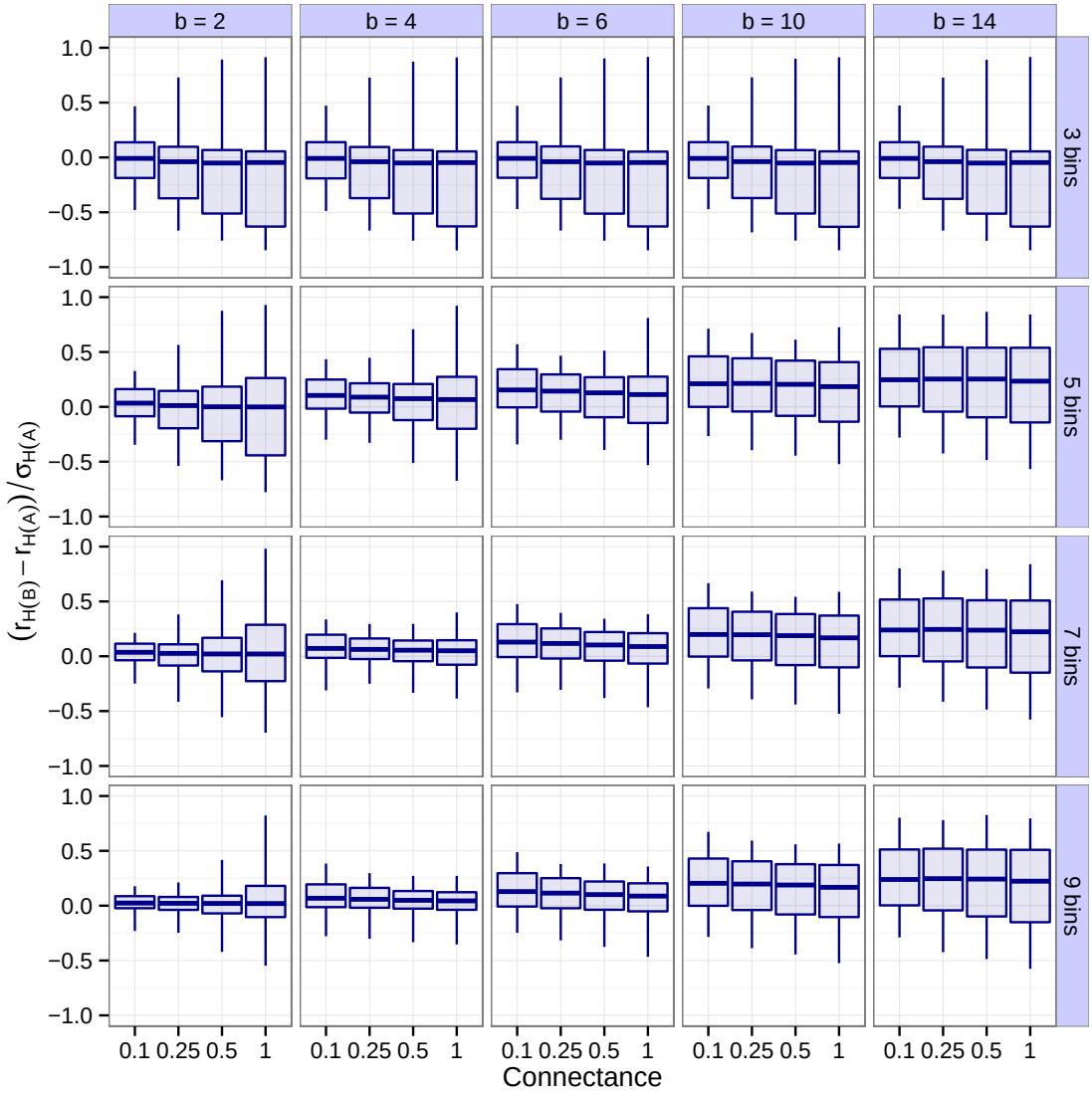

Figure S17: As Fig. S11, but with the number of species fixed at $S = 250$.

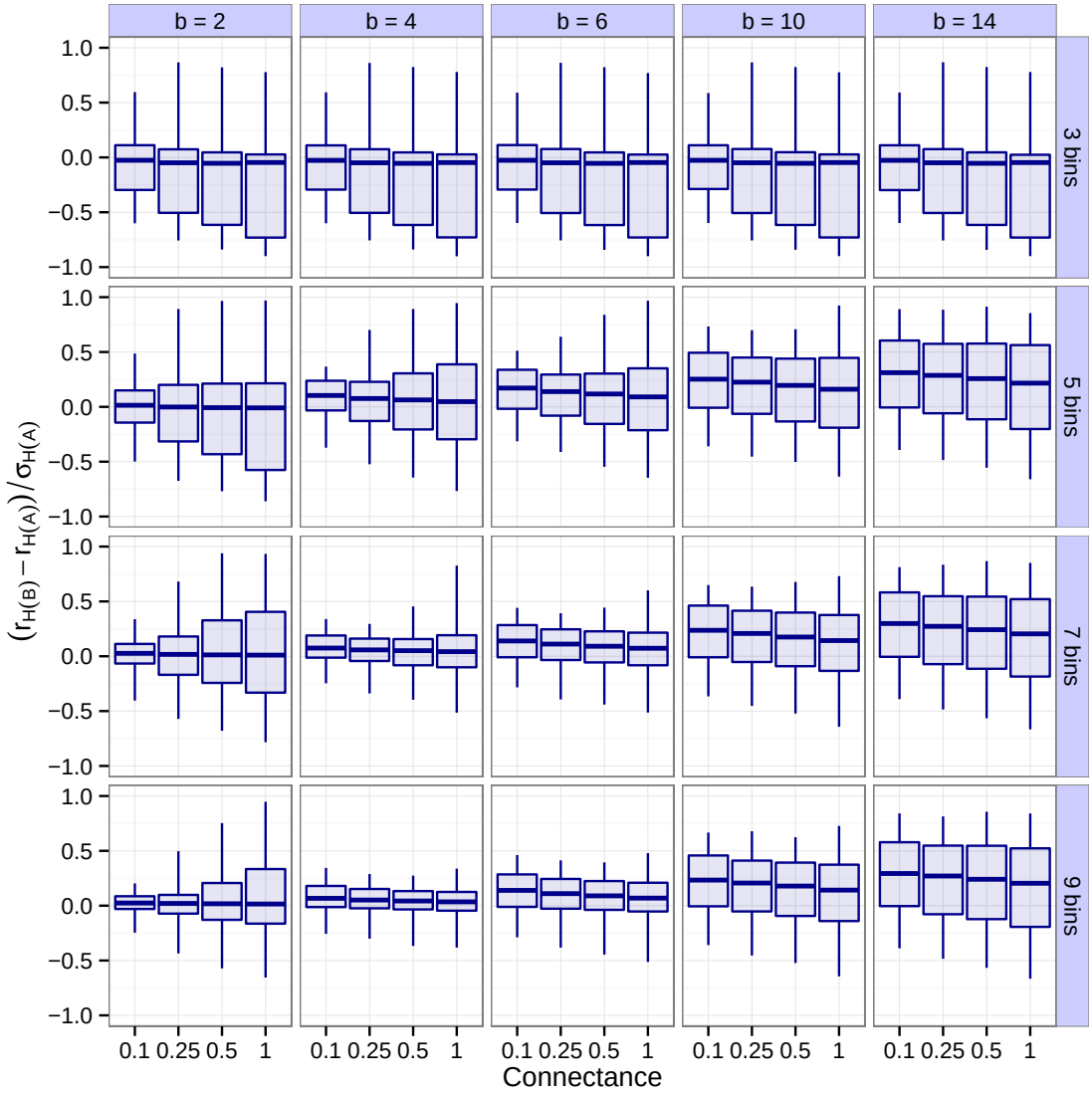

Figure S18: As Fig. S11, but with the number of species fixed at $S = 500$.

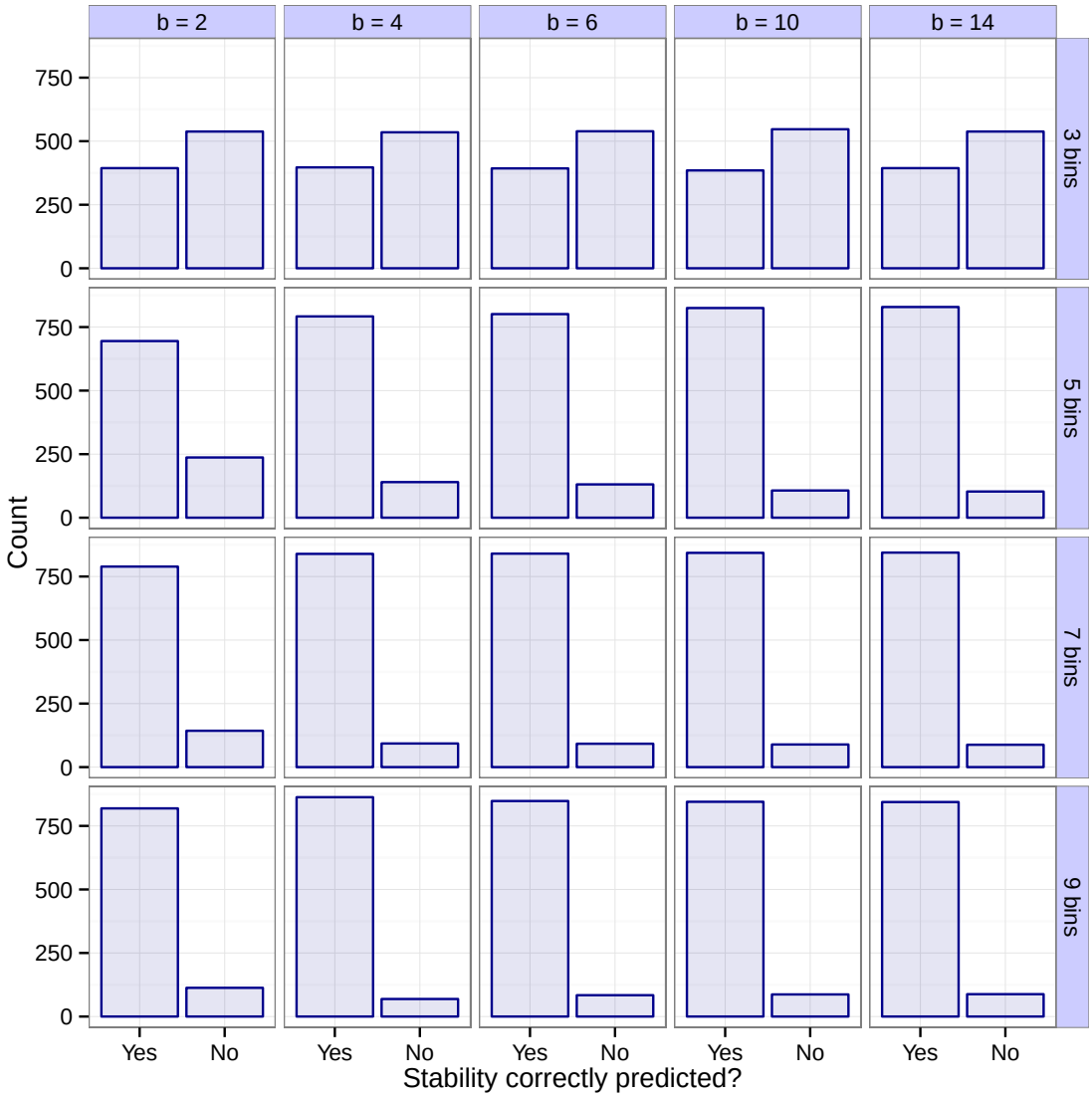

Figure S19: The number of cases in which stability is correctly predicted, out of those matrices whose leading eigenvalues have a magnitude less than one twentieth of the total spread of the real parts of all eigenvalues: $|r_A|/\sigma_A < 0.05$. Rows correspond to different values of the binning resolution b; columns to different numbers of bins k. The rate of misclassification is 10%.

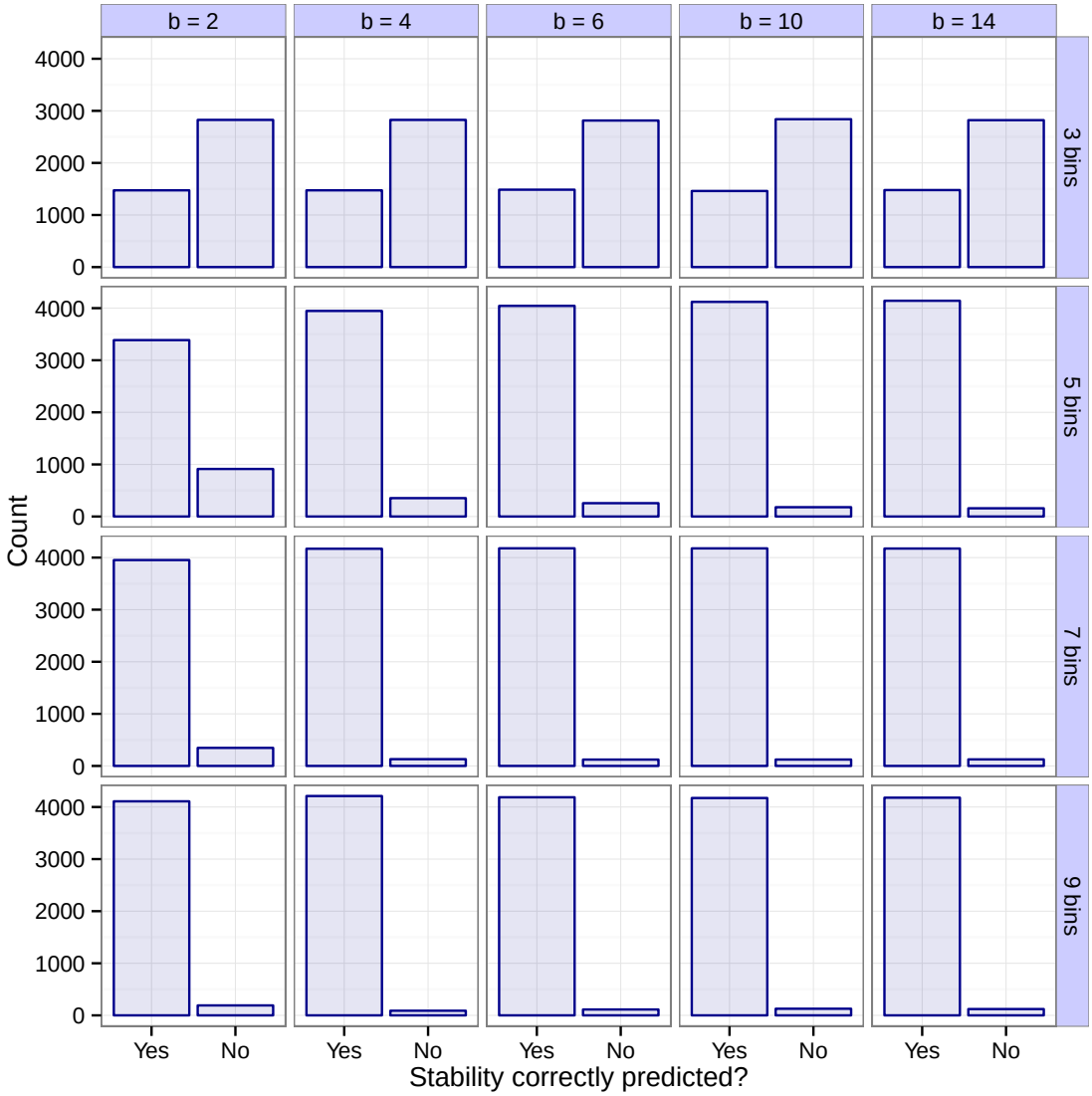

Figure S20: As Fig. S19, but for all those matrices for which $|r_A|/\sigma_A < 0.1$.

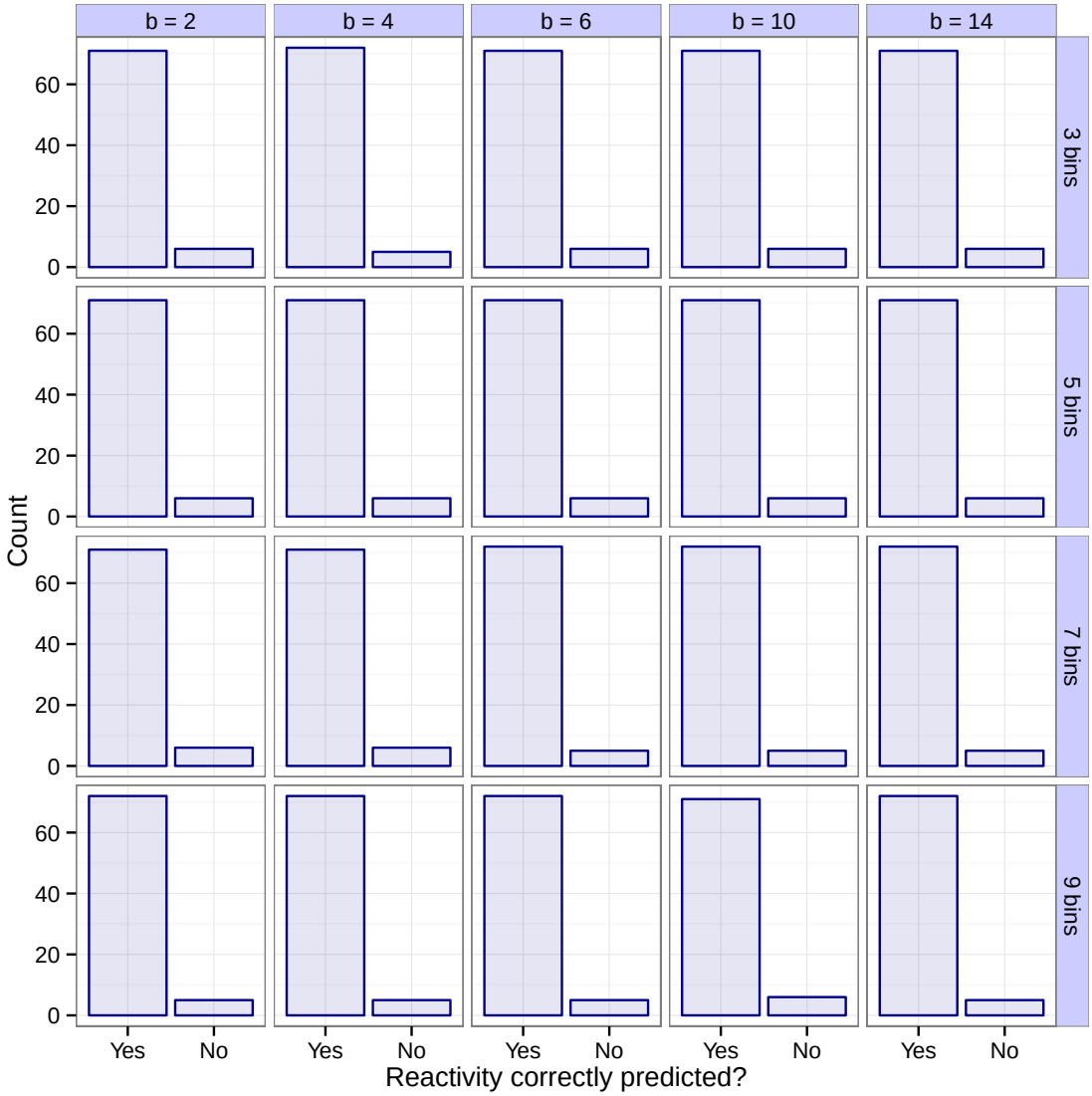

Figure S21: As Fig. S19, but for reactivity instead of stability.

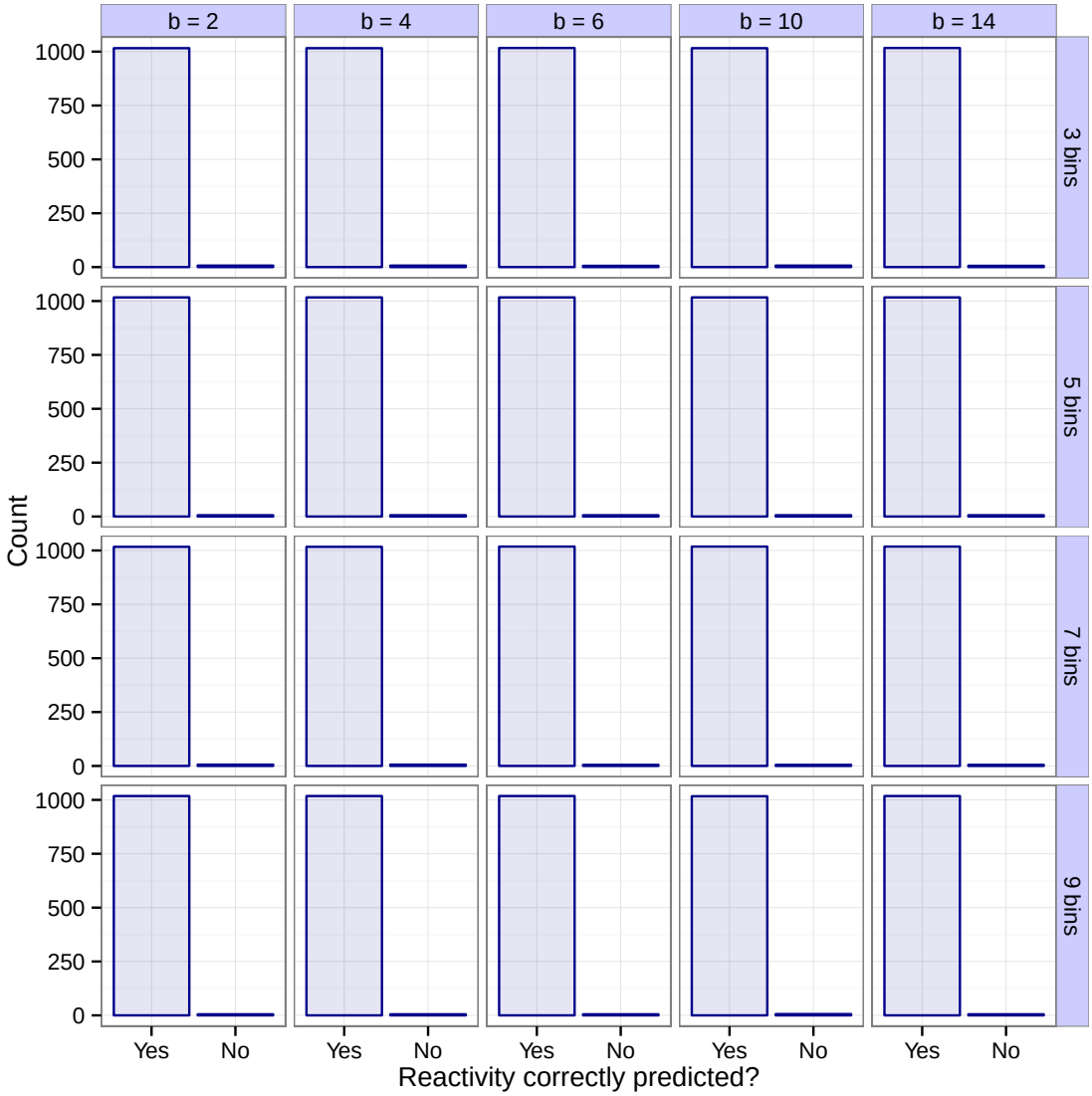

Figure S22: As Fig. S20, but for reactivity instead of stability.

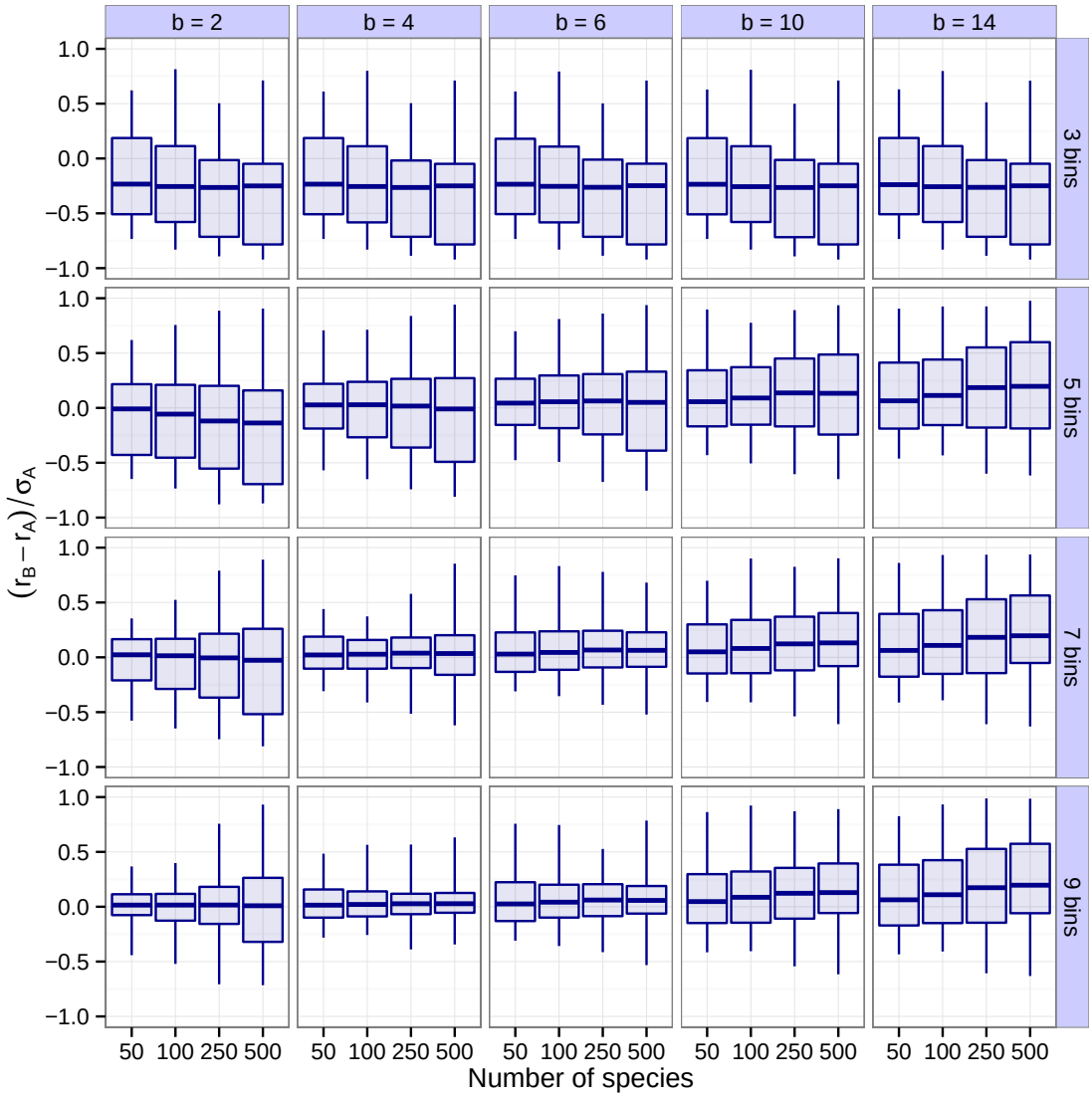

Figure S23: As Fig. S1, but with a misclassification rate of 10%, and all interaction strengths drawn from lognormal distributions of varying means and variances (see main text).

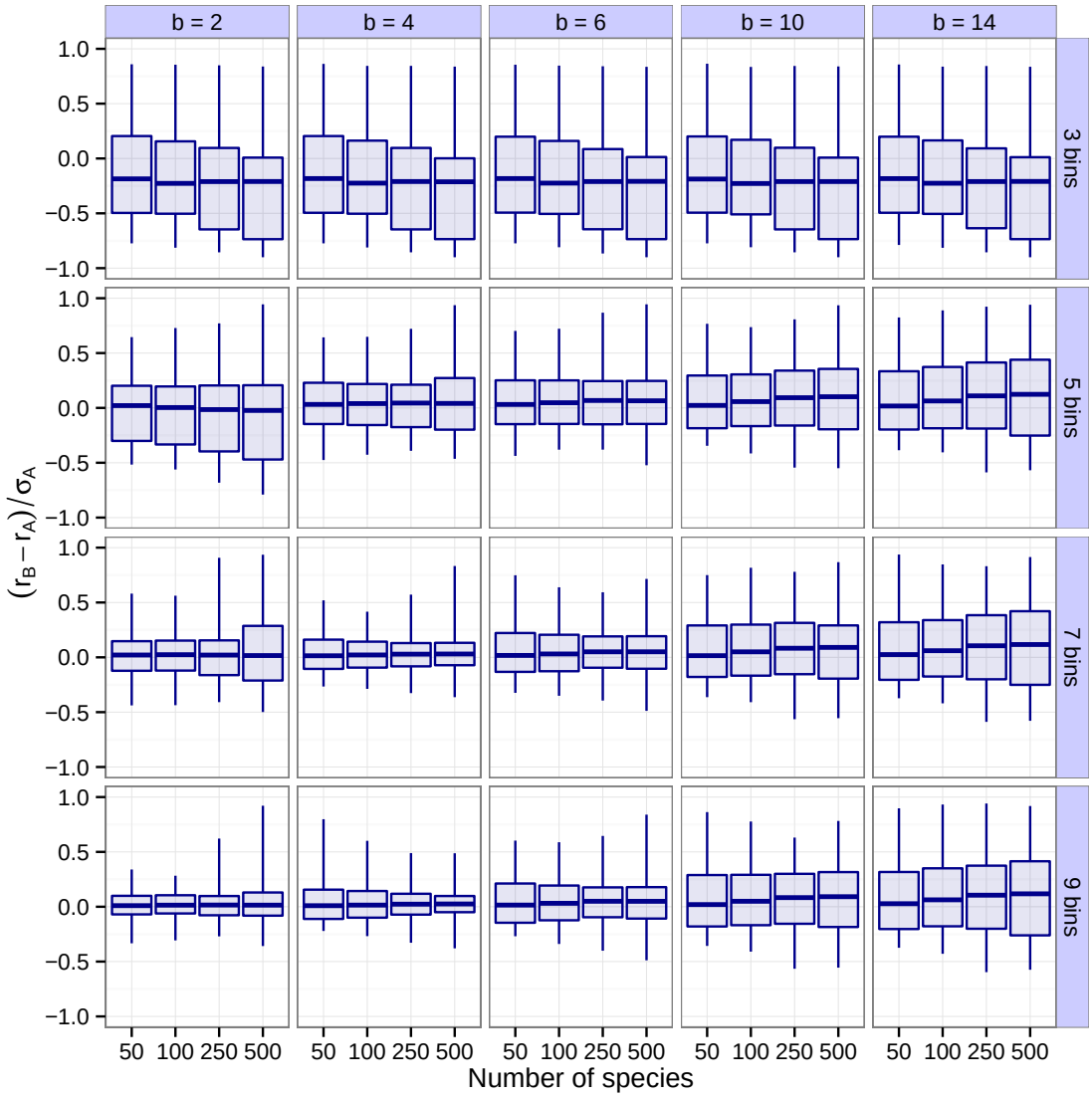

Figure S24: As Fig. S23, but with all interaction strengths drawn from variously parameterized Gamma distributions.

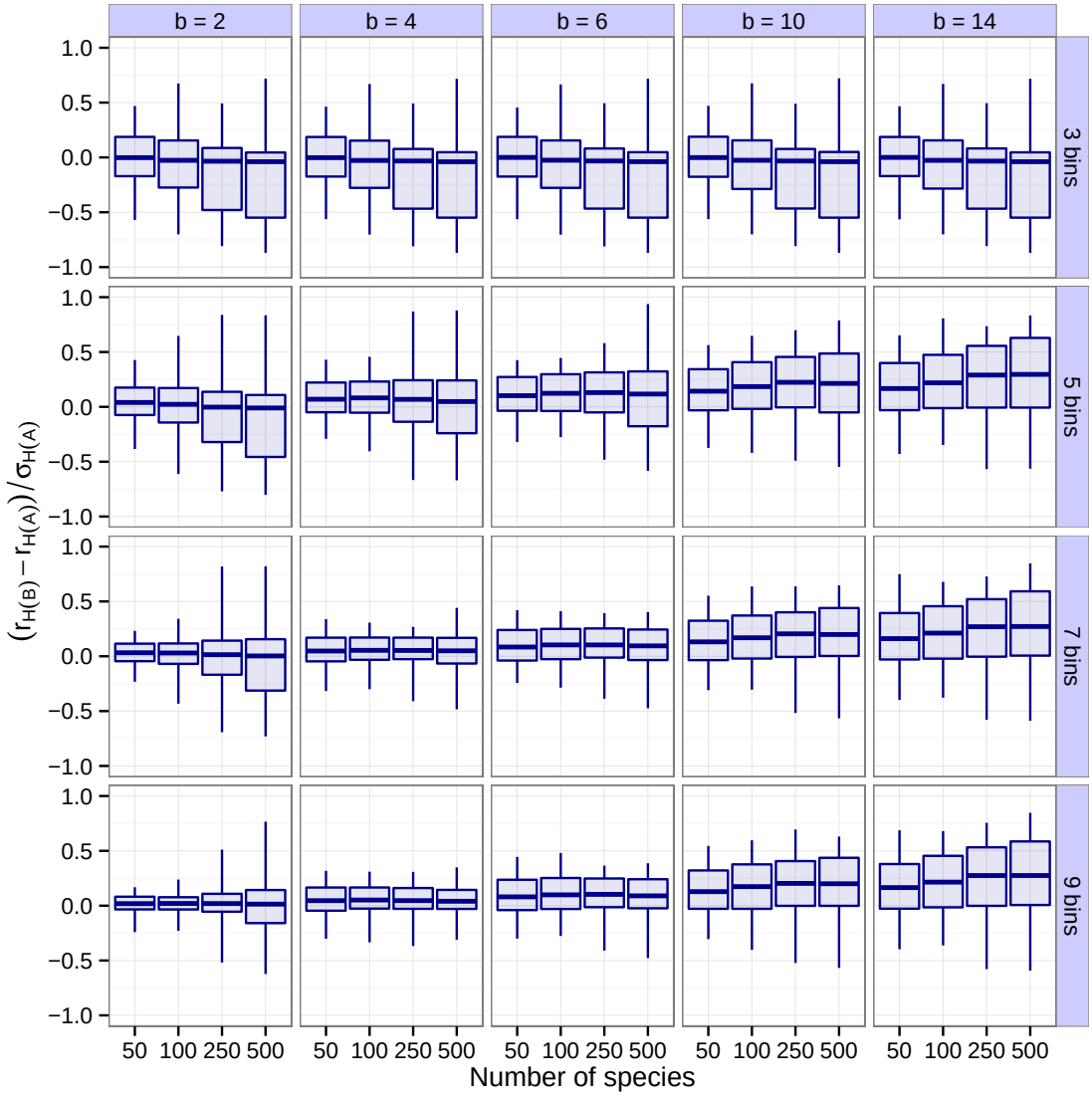

Figure S25: As Fig. S23, but for reactivity instead of stability.

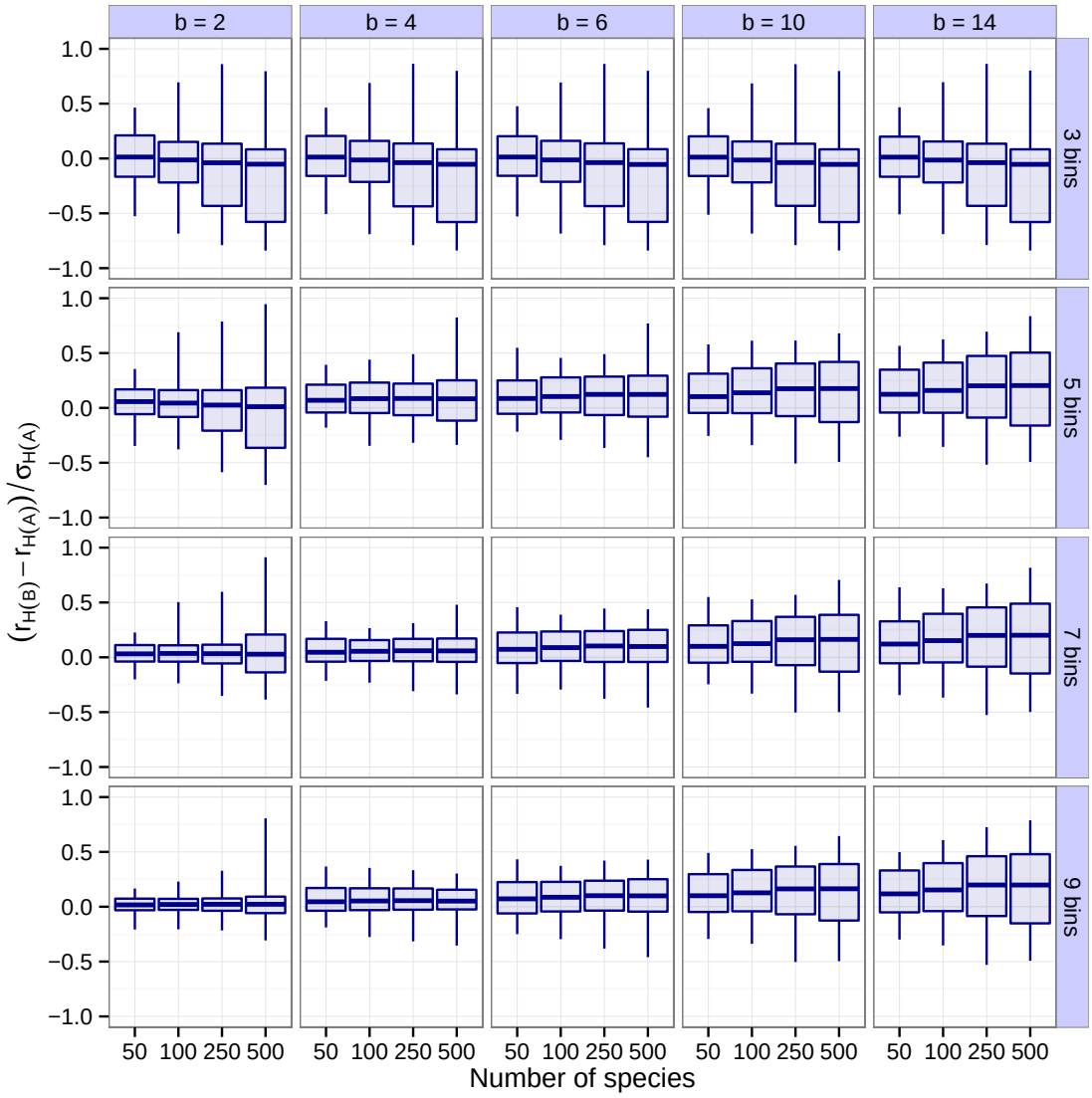

Figure S26: As Fig. S25, but with all interaction strengths drawn from variously parameterized Gamma distributions.

References

Allesina, S., Tang, S., 2012. Stability criteria for complex ecosystems. Nature 483, 205–208.

Berlow, E. L., Dunne, J. A., Martinez, N. D., Stark, P. B., Williams, R. J., Brose, U., 2009. Simple prediction of interaction strengths in complex food webs. Proceedings of the National Academy of Sciences USA 106, 187–191.

Edelman, A., 1988. Eigenvalues and condition numbers of random matrices. SIAM Journal on Matrix Analysis and Applications 9, 543–560.

Ginibre, J., 1965. Statistical ensembles of complex, quaternion, and real matrices. Journal of Mathematical Physics 6, 440–449.

Girko, V. L., 1984. The circle law. Theory of Probability and its Applications 29, 694–706.

Henrici, P., 1962. Bounds for iterates, inverses, spectral variation and fields of values of non-normal matrices. Numerische Mathematik 4, 24–40.

Lee, S. L., 1996. Best available bounds for departure from normality. SIAM Journal on Matrix Analysis and Applications 17, 984–991.

Marčenko, V. A., Pastur, L. A., 1967. Distribution of eigenvalues for some sets of random matrices. Math USSR Sbornik 1, 457–483.

R Development Core Team, 2008. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Sommers, H. J., Crisanti, A., Sompolinsky, H., Stein, Y., 1998. Spectrum of large random asymmetric matrices. Physical Review Letters 60, 1895–1898.

Tang, S., Pawar, S., Allesina, S., 2014. Correlation between interaction strengths drives stability in large ecological networks. Ecology Letters 17, 1094–1100.

Tao, T., Vu, V., Krishnapur, M., 2010. Random matrices: Universality of esds and the circular law. Annals of Probability 38 (5), 2023–2065.

Trefethen, L. N., Embree, M., 2005. Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators. Princeton University Press, New Jersey, USA.

Williams, R. J., Martinez, N. D., 2000. Simple rules yield complex food webs. Nature 404, 180–183.

Wolfram Research Inc., 2014. Mathematica, version 10.0. Wolfram, Champaign.