1 Code/introduction

All the code that we use in the paper, and in the appendix, is available as an R package. It can be installed via

```
remotes::install_github("bbolker/betararef")
```

You will need to install the `remotes` package from any standard CRAN repository first (if you have the `devtools` package installed, you can use that instead). Since the package contains only R code (no compiled C/C++/FORTRAN code), you should *not* need to have other development tools installed.

To begin using the package you need to load it (it needs to be installed only once per R installation, but loaded every time you start a new R session):

```
library(betararef)
```

2 Community simulator

The simulation first simulates a set of 'endemic' species in a specified number of patches, then mixes some proportion of each species randomly across all patches according to specified rules. The parameters of the simulation are as follows:

n.abund number of abundance categories (e.g. 2={common, rare}; 3={common, intermediate, rare})

p.abund proportional change between abundance categories (rank-abundance curve is geometric: overall frequencies of abundance categories are proportional to $\{1, \phi, \phi^2, \ldots\}$)

spcat number of species per site per abundance category (i.e. the total number of (endemic) species per site is `spcat*n.abund`, and the complete size of the species pool is `spcat*n.abund*n.site`): may be < 1, but the total number of species per abundance category (`spcat*n.site`) must be an integer

n.site number of sites

n.indiv.site number of individuals per site

p.mix vector of mixing probabilities for each abundance category: 0=no mixing, 1=complete mixing, e.g. when `n.abund=2`, {0,1} means that common species are completely mixed and rare species are completely endemic

rand "none": *expected* number of individuals per site/species is reported; "multinomial": number of individuals per site is held constant, proportions chosen randomly; "poisson": number of individuals per site/species is Poisson distributed

In general the simulator returns a site $\times$ species matrix, which we can post-process with the tools from `vegan` to compute beta diversities.

Here are some typical examples. If we use all of the default values for `betasim`, we get 3 sites (rows) with a total of 30 species: 5 abundance classes $\times$ 2 species per site $\times$ 3 sites. By default there is no mixing among sites (100% beta diversity), so looking at the default plot (rows=sites, columns=species) we see that each site has blocks of 2 species from each abundance class (grayscale indicates prevalence) (Figure 1). (The default for `betasim` (but not for `simComm`) is to introduce

no randomness at all, simply returning a matrix with the *expected* number of individuals of each species at each site.)

If we allow 100% mixing, then all species are present at all sites in the same numbers (Figure 2). We can also allow partial mixing (Figure 3) and Poisson variation in the output (Figure 4).

We can change the number sites or the number of abundance classes: by default, the `spcat` (number of species per site per abundance category) will be adapted to try to make the total number of species come out correctly (i.e. `spcat` will be set to `totsp/(n.site*n.abund)`: Figure 5). If `totsp` is not an even multiple of `n.site*n.abund`, the results may be slightly unpredictable ...

`calcbeta` calculates distances from the centroid (by default, using a Jaccard index on presence-absence data: Figure 6).

`simComm` simulates multiple communities (with possibly different characteristics) at the same time (Figure 7).

3 Alternative diversity indices

In the main text we focus on the Jaccard index based on binary (presence-absence) data. The literature is full of alternative diversity measures; while our analysis focuses on the effect of sampling rather and hopes to be generally applicable across different indices, we should at least consider what would happen if we were to use an index based on continuous (count) data instead. Figure 8 shows results for the Manhattan index (we use either Hellinger or square root transformations as recommended by **legendre˙2013**; however, **legendre˙2013** have also criticized the Manhattan index, saying that in particular it does not properly handle situations where multiple sites are missing the same pairs of species).

This analysis extends the general principle that we can understand the effects of sampling on beta diversity by considering the expected effect of sampling intensity on variance. In this case, the responses are counts rather than binomial (binary) responses, so the expected variance of the response, and thus the estimated beta diversity, is proportional to $1/\sqrt{N}$. In contrast to the binary Jaccard results, there is no hint of non-monotonicity on the beta diversity curves. The number of sites has negligible effects on the computed beta diversity: the rank-abundance curve has moderate effects, although they disappear in extreme (low- or high-mixing) cases.

chase˙2011 have recommended the Raup-Crick dissimilarity index, which characterizes diversity as the probability (based on permutation tests) that two communities have the same species composition. Raup-Crick indices are robust to many different aspects of community structure, and indeed when we run the same set of analyses as Figure 3 in the main text, we see that this index is robust to variation in all axes of ecological variation (number of sites, rank-abundance curve, mixing probabilities) *except* for sampling intensity/local population size (Figure 9). In other words, the figure shows the same general decrease in beta diversity with increasing sample size as other density-based metrics such as the Manhattan index, but the curves are identical across all other axes of variation.

4 Power analyses

We ran power analyses of hierarchical rarefaction analysis, varying the true difference in beta diversity, the average population/sample size per patch, the difference in population/sample size per patch, and the difference in the number of patches. We kept the average number of patches equal to 20, and the number of species per community equal to 20.

The results show that for a per-patch population size of 20, the test is actually slightly conservative (i.e., the probability of rejecting the null hypothesis with $\alpha = 0.05$ when it is true (pmixdiff=0) is actually slightly less than 0.05; for a per-patch population size of 10, the results are slightly anti-conservative. The power curves (i.e., power to detect a specified level of difference in beta diversity)

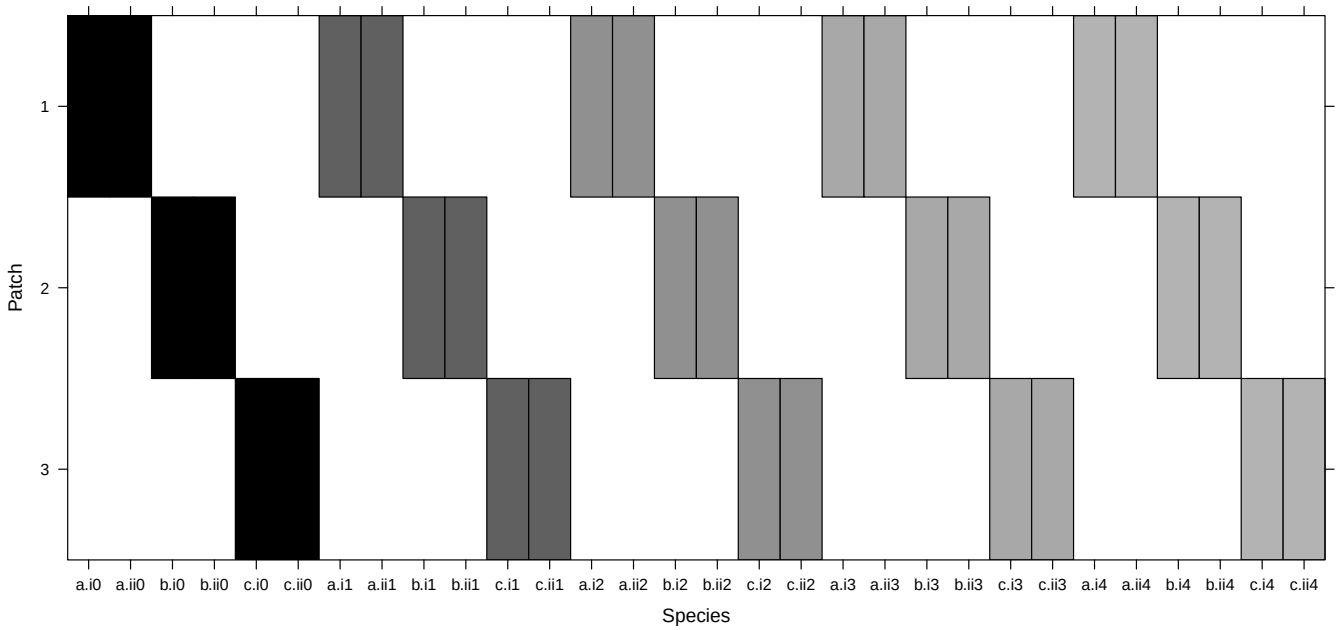

Figure 1: Species matrix for default `betasim` settings: 3 sites, 10 species per site (5 abundance classes/2 spp per abundance class), no mixing.

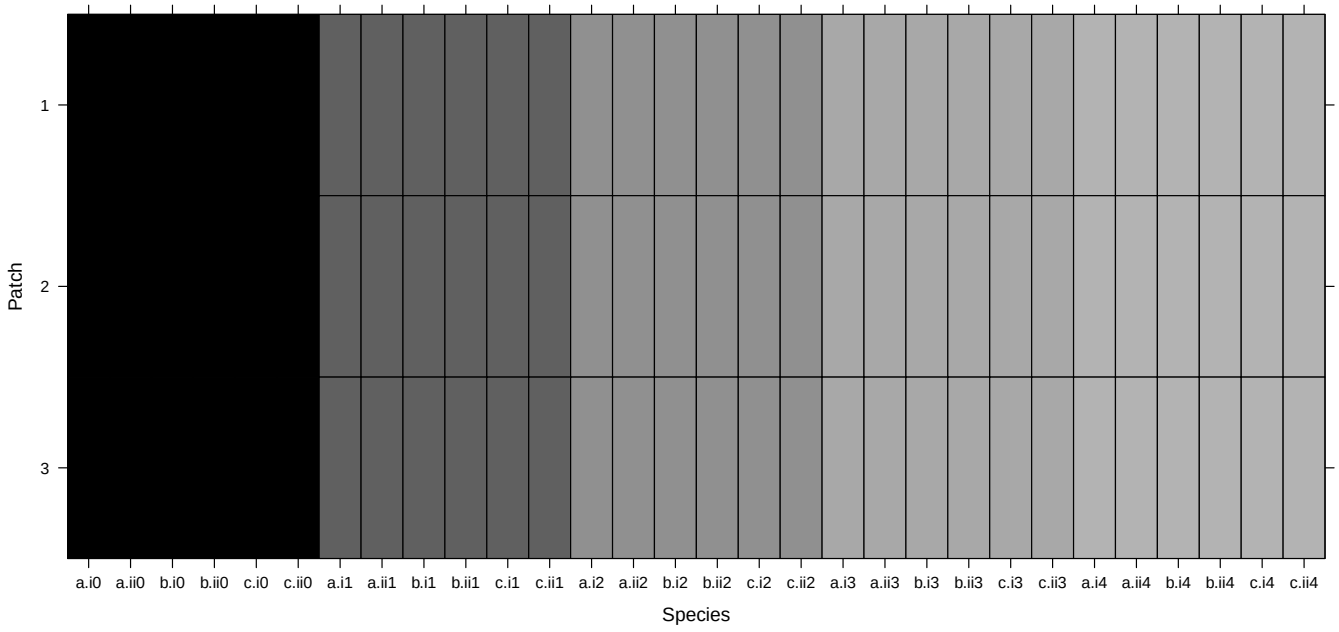

Figure 2: Species matrix for `betasim()` with default parameters (3 sites/5 abundance classes/2 spp per AC) but 100% mixing.

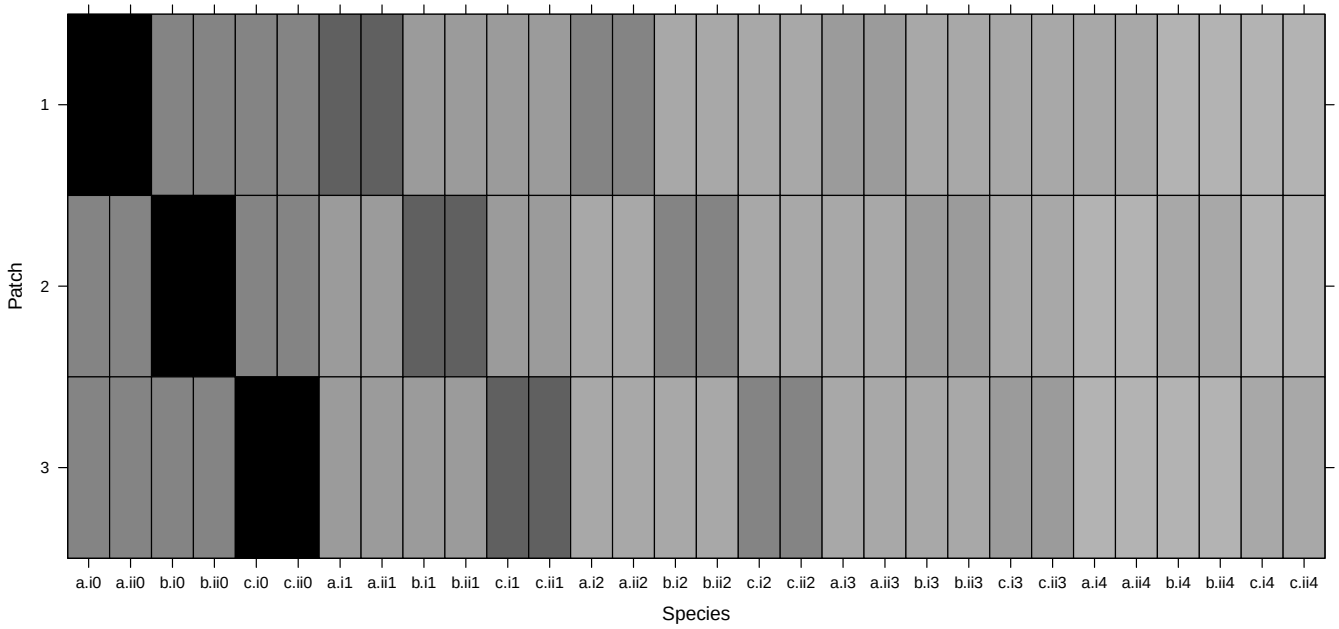

Figure 3: Species matrix for `betasim()` with default parameters (3 sites/5 abundance classes/2 spp per AC) and 50% mixing for all abundance classes.

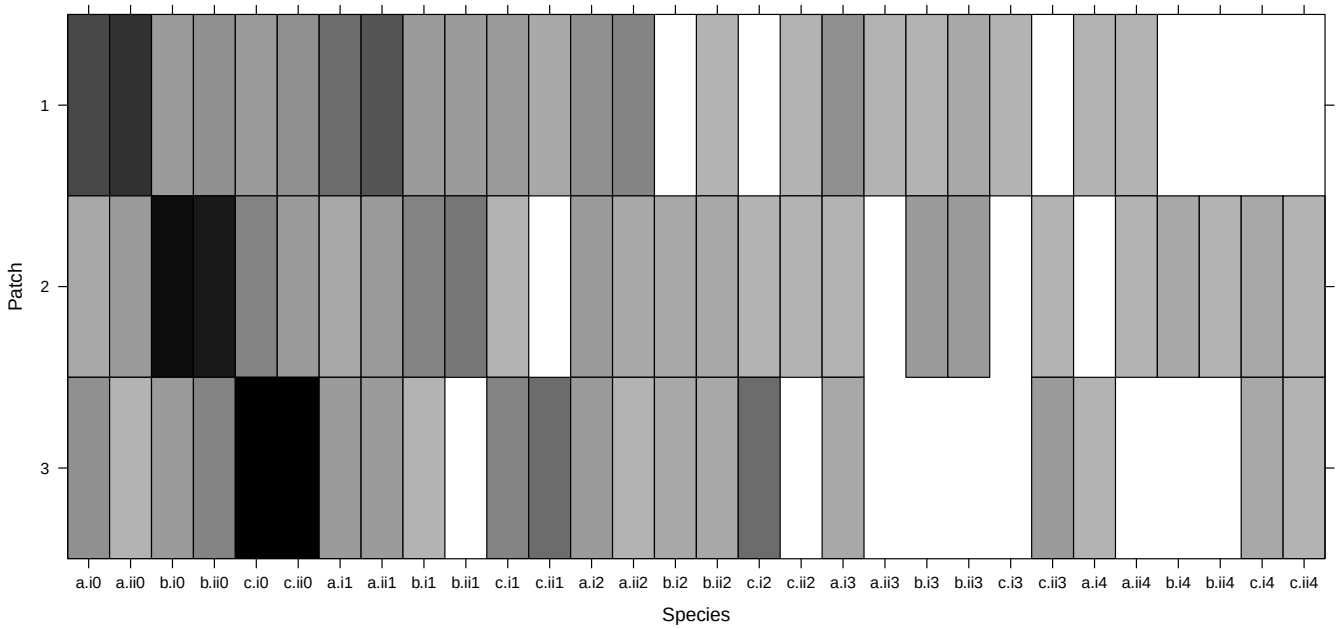

Figure 4: Species matrix for `betasim()` with default parameters (3 sites/5 abundance classes/2 spp per AC) with 50% mixing for all abundance classes and incorporating Poisson variation in the output.

change little as a function of per-patch population size or population size difference, and declines slightly with increasing asymmetry in the number of patches per site, probably driven by low power in the community with the small number of patches.

Mean and standard error of type I error rate (true difference in asymptotic beta diversity/mixing parameter=0), by differences in numbers of sites, *not* including patch size=20 case for 30-site {5,35} case:

sizediff	a.mean	a.se
0	0.009	0.003
20	0.015	0.003
30	0.059	0.004

For patch size=20, number of sites={5,35}:

a.mean	a.se
0.133	0.009

5 Sample-based rarefaction

As we mention in the paper, our approach to multi-level rarefaction is based on individual-level rarefaction, in part because of the commonness of community data with only a single sample per patch (in which case sample-based rarefaction is useless) and in part because of the complexity of bookkeeping in the multi-sample/sample-based rarefaction case.

If we stop worrying about the constraints of rarefying each patch to the minimum number of individuals found in any patch, in any treatment (which is what necessitates the complex matching strategy implemented in the package), we can come up with a reasonably simple strategy for rarefaction.

- pick the per-patch size N to rarefy to
- for each patch in each community,
  - while the accumulated patch size is $< N$, pick a sample from that patch without replacement
  - if the next-chosen patch makes the patch size $> N$, either
    - take only the number of individuals from the sample that brings the patch up to the target, *or*
    - add the sample to the patch community, then remove enough individuals from the community at random to bring it back down to the target size

We have not implemented this functionality yet.

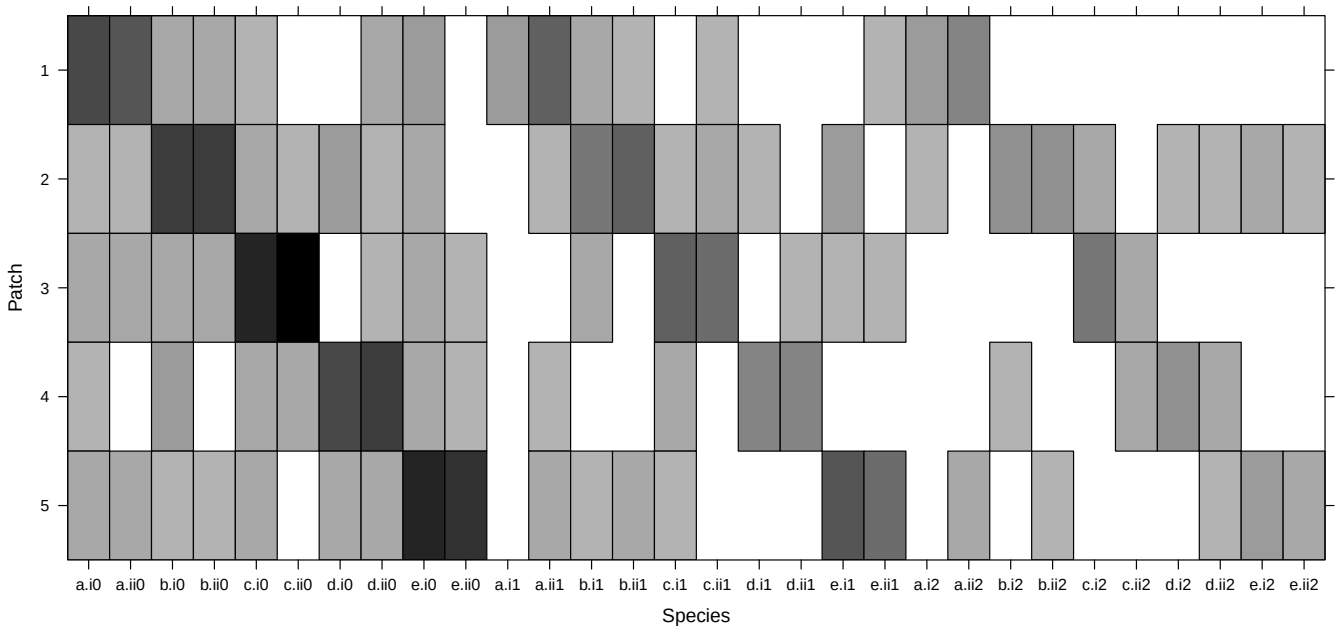

Figure 5: Species matrix for `betasim()` with 5 sites, 3 abundance classes, 2 spp per AC, 25% mixing for all abundance classes, and Poisson variation.

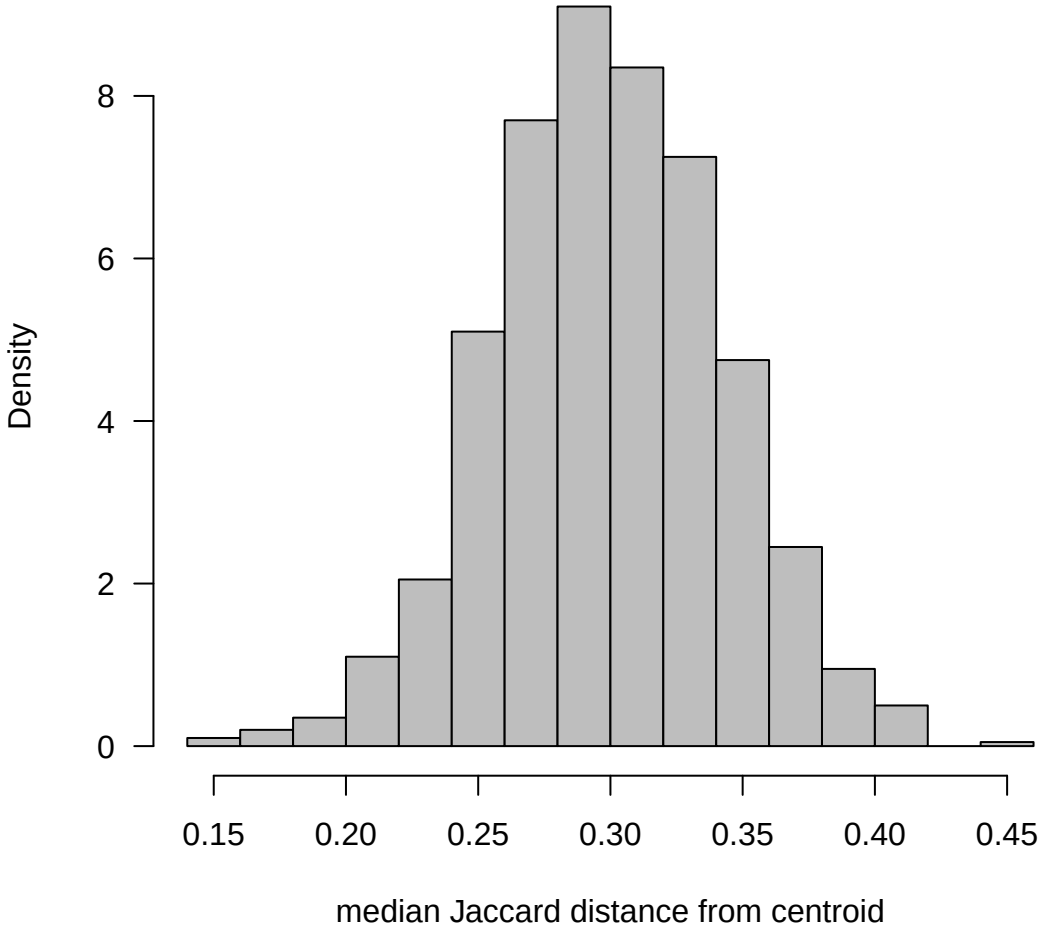

Figure 6: Histogram of `calcbeta` output for 10000 simulated communities (median Jaccard distance to centroid).

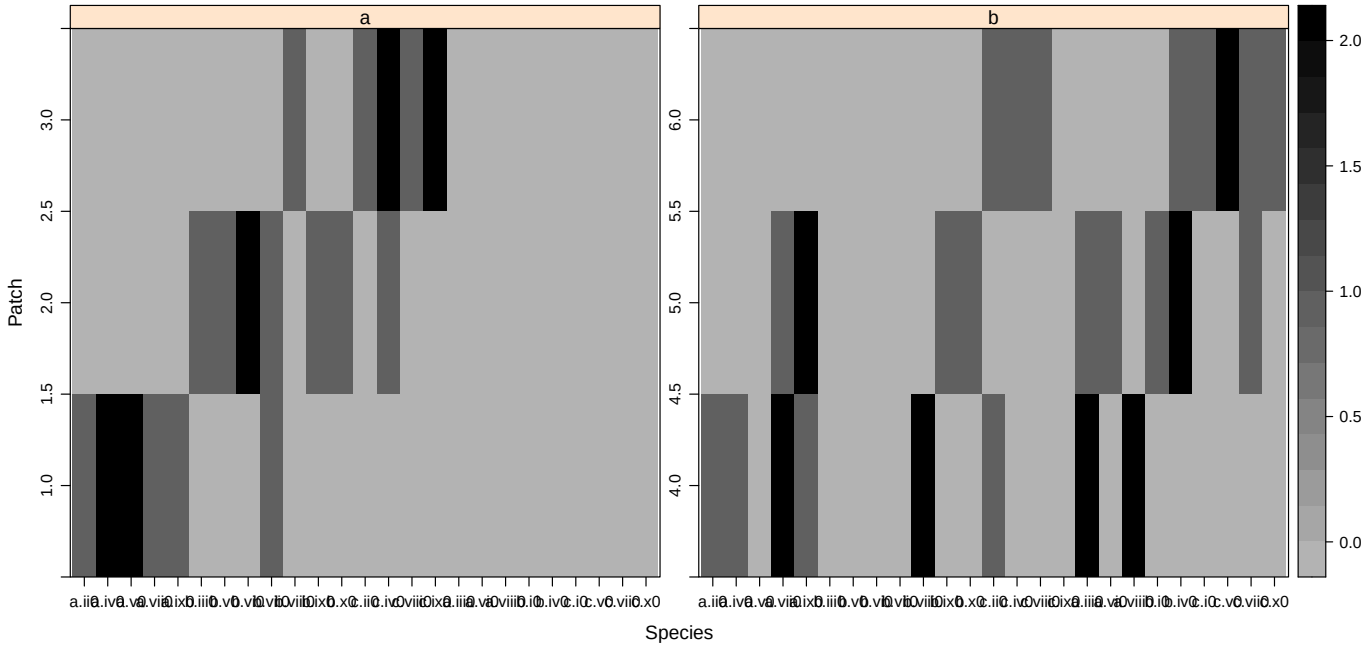

Figure 7: Output of `simComm`: two communities simulated simultaneously, with 30 species at each site and mixing proportions of 0.2 and 0.7 respectively.

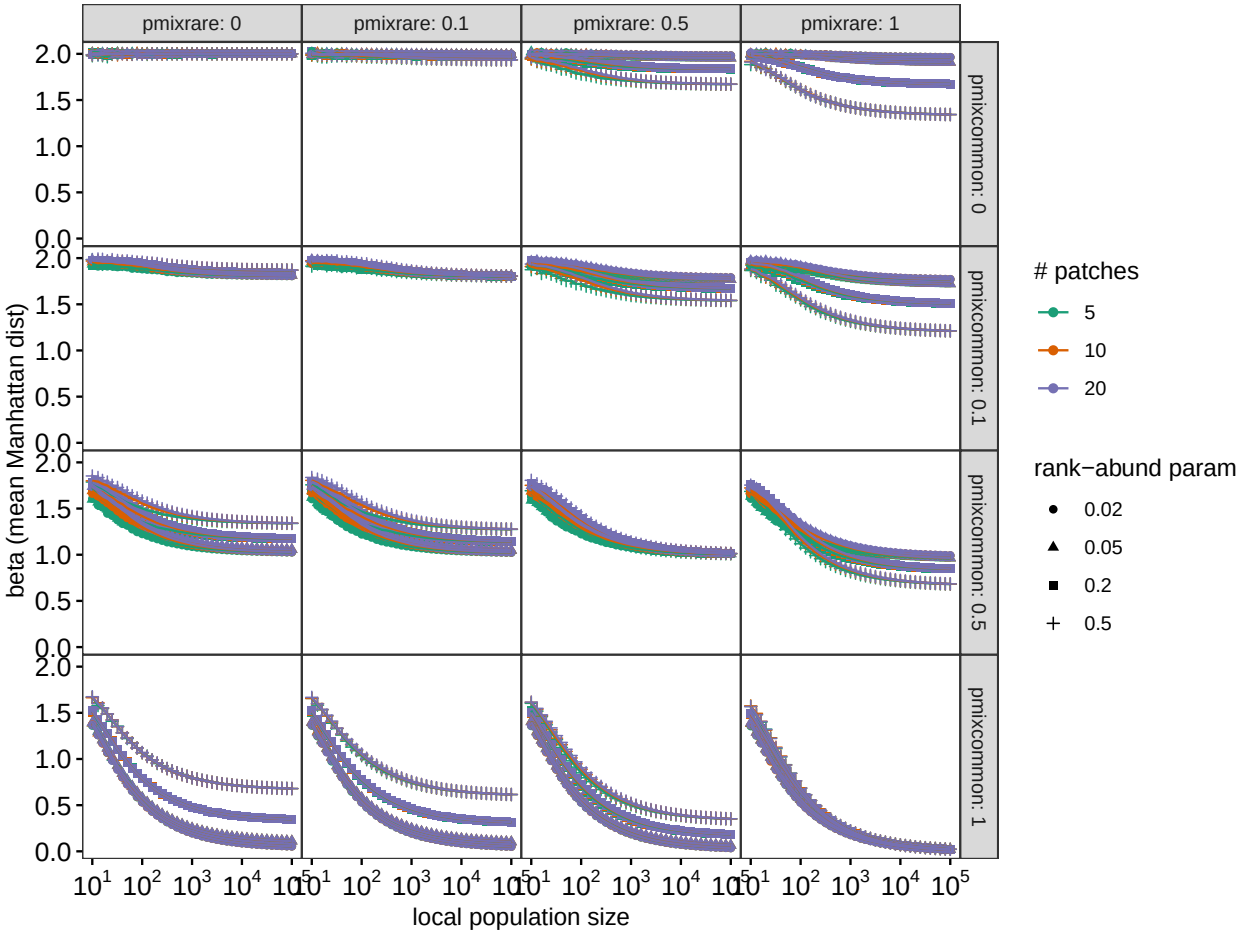

Figure 8: Diversity results for Manhattan index

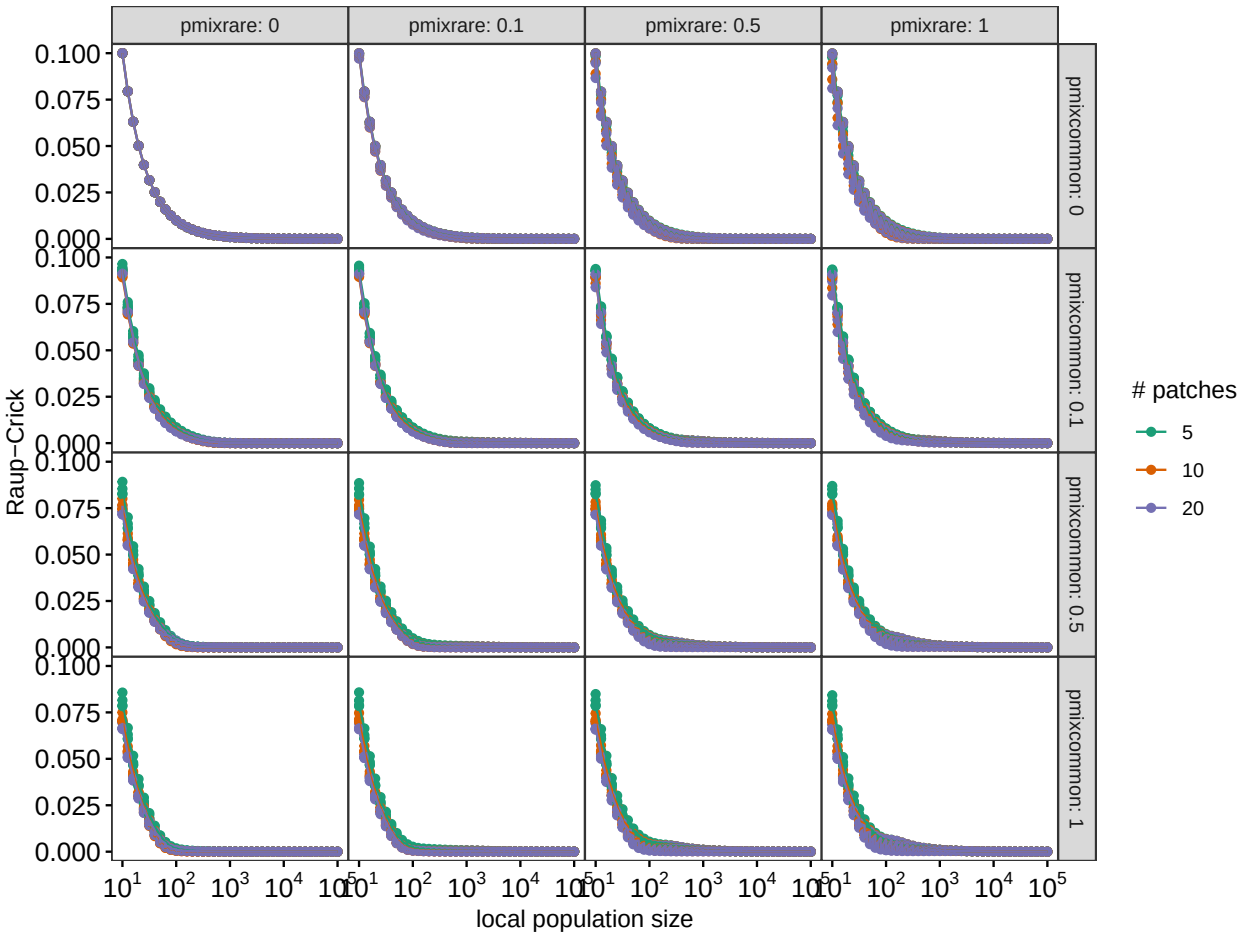

Figure 9: Diversity results for Raup-Crick index

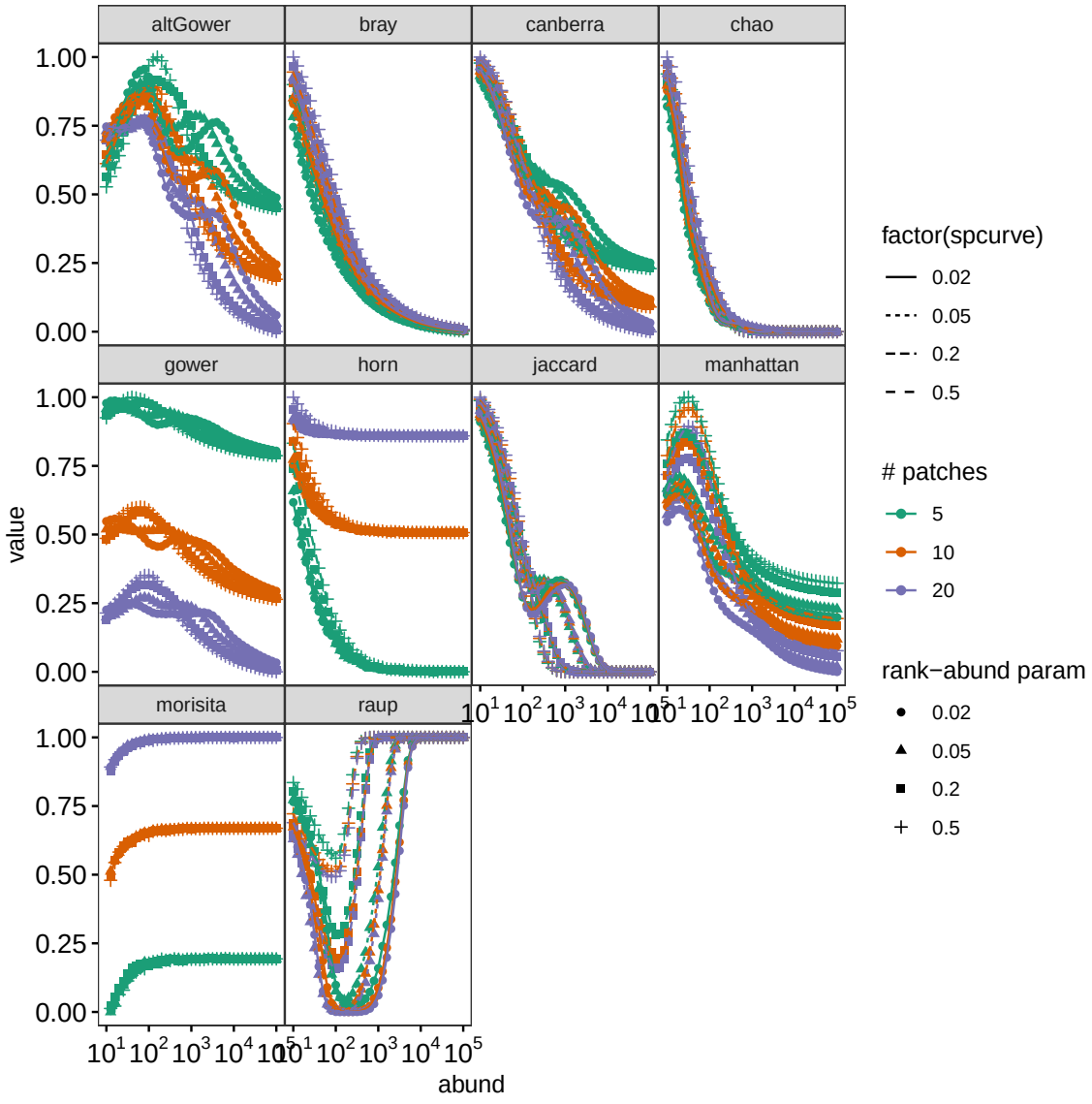

Figure 10: Beta diversity curves (scaled 0-1 for each diversity metric) as a function of sample size for a range of distribution for `pmixrare`=`pmixcommon`=0.5.

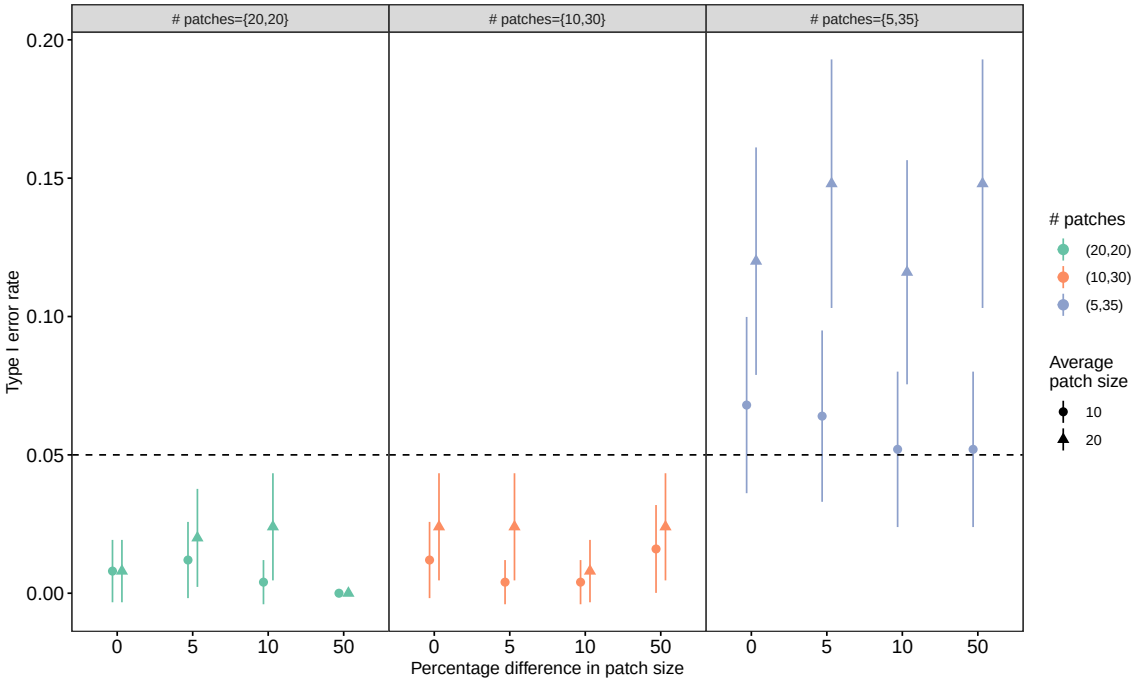

Figure 11: Type I error rates (rejection rates when `pmixdiff`=0).