Appendix 2

More Efficient Proposals Over Correlation Matrices

1 Motivation

Correlation matrices are a special case of covariance matrix. They have all the latter's properties, including symmetry and positive semi-definiteness (PSD), in addition to the requirement of a unit diagonal. In Bayesian inferential contexts involving covariance matrices, it is common to factor covariance matrix C into the matrix product SRS, where S is a diagonal matrix of standard deviations and R is a correlation matrix. Priors can then be specified on these components separately. Under the phylogenetic multivariate Brownian motion model, the rate matrix is a covariance matrix, describing the shape of the multivariate normal distribution — scaled multiplicatively by the branch length — from which displacements to the location of some vector of traits is drawn.

A problem arises when making proposals to correlation matrices during the Metropolis-Hastings algorithm (**?**) used to numerically approximate the Bayesian joint posterior distribution of model parameters. The problem involves the PSD constraint, which with increasing matrix dimensionality rapidly narrows the window of viability available to each correlation coefficient, conditional on the values observed at all other positions of the correlation matrix. This is implied by the narrowing marginal distribution of each coefficient in samples from an LKJ distribution, which can be described by a beta distribution stretched into the (-1,1) range with shape parameters $\eta - 1 + D/2$, where D is the dimension of the correlation matrix. It can also be demonstrated experimentally by sampling a correlation matrix from a flat LKJ distribution ($\eta = 1$), selecting one of its off-diagonal elements, and identifying the magnitude of the interval about that element for which the PSD condition is preserved. Averaging across many replicates, we obtain an estimate of the expected interval over which proposals to individual elements of a correlation matrix may be made for matrices of a given size, which we summarize graphically (Figure **??**).

As this interval shrinks, the Metropolis-Hastings algorithm slows tremendously with respect to its mixture over the posterior distribution of correlation matrices if naive proposal distributions are used that do not respect the constraint of positive semi-definiteness. Additionally, the number of parameters to be estimated increases quadratically with the dimension of the correlation matrix, as a correlation matrix of dimension n has (n choose 2) off-diagonal elements. And tests for positive-semidefiniteness — for example, involving eigendecomposition and the surveyance of positive eigenvalues — can themselves become quite computationally cumbersome, especially at high dimensionality. As such, approaches that rely on the rejection of invalid proposals must either sample from more and more conservative distributions over increasingly many elements of the correlation matrix, or else make tremendously more proposals over the course of the MCMC.

Here, we describe a proposal distribution over correlation matrices that makes proposals to multiple correlation parameters simultaneously and is guaranteed to sample from the PSD space. It is asymmetric, and our description therefore also entails the calculation of forward and backward proposal probabilities. In addition, it is tunable on a per-trait-index basis, allowing for the optimization of acceptance probabilities contingent upon the narrowness of the target distribution. Finally, we describe its implementation with respect to the Cholesky factor of the correlation matrix, which is a commonly used internal representation of a correlation matrix for the purpose of computational efficiency (e.g. in transforming tip characters or specifying non-centered parameterizations in Gaussian Process Regression). The proposal distribution also samples a new Cholesky factor in $O(n^2)$ time complexity, removing the need to perform a more costly $O(n^3)$ Cholesky factorization. One minor complication of the below described proposal distribution involves its return of a Cholesky factor corresponding to a permuted correlation matrix, rather than one with the original row and column indexing. In our implementation, an optional argument can be toggled to return to the original order at the cost of some of the aforementioned efficiency. Instead, we prefer to permute the (e.g. multivariate normal) data whose density is being calculated, a much more efficient procedure that nevertheless requires slightly more bookkeeping.

Following our description of the proposal distribution, we conduct a short validation of its performance, first using it to sample via the Metropolis-Hastings algorithm from a flat LKJ distribution — uniform over all correlation matrices — and comparing these samples to those drawn directly from the LKJ. Then, we use the proposal distribution to approximate a more informative distribution: the posterior distribution of the correlation matrix of a 10-dimensional multivariate normal random variable whose means and variances are known, conditional on 50 samples from that distribution and an LKJ($\eta = 1$) prior. We compare these to the same distribution independently approximated through more innocuous means: a uniform sliding window proposal over all choose(10, 2) pairwise elements of the correlation matrix. The former is run for 1E7 iterations and the latter for 5E7, with thinning occurring at an interval of 1E3 and 5E3, respectively. The first 20% of each chain is discarded as burn-in, and automatic tuning is performed across 50 sub-rounds of burn-in to target a per-trait (in our novel proposal distribution) and per-correlation (in the uniform sliding window proposal distribution) acceptance probability of 0.234 (**?**). In practice, this resulted in acceptance probabilities ultimately falling in the interval (0.20, 0.24).

2 DESCRIPTION

The proposal distribution was inspired by the *extended onion method* of the **?** paper by Lewandowski, Kurowicka, and Joe for sampling from the eponymous LKJ distribution. In this algorithm, the Cholesky factor of a correlation matrix is built iteratively up to the desired dimensionality n, one layer at a time, as one might grow an onion. Our proposal distribution simply unravels the last step of the extended onion method, corresponding to the construction of the correlations of the nth variable with the remaining n - 1 variables of the n x n correlation matrix. In the extended onion method, this final column of the upper Cholesky factor U is constructed through sampling a single beta distributed random variable y from a beta($n/2$, η), and a vector $u = (u_1, ..., u_{n\text{-}1})$ uniformly sampled from the surface of an n-dimensional hypersphere. The u are then rescaled by the square root of y to form the first n - 1 elements of the column of U. The square root of (1-y) then forms the last entry of that column, ensuring unit length.

Working backwards, we can straightforwardly identify the unique y and u necessary to produce some current U, conditional also on its (n - 1) x (n - 1) submatrix. We can then resample these values centered on these targeted states — disallowing any room around them will reproduce U, and resampling from $y \sim$ beta($n/2$, η) and $u \sim$ uniform on the unit hypersphere will generate a sample from the conditional LKJ(η). Between these extremes, we can resample y' from within a window of tunable width w centered around the target y in proportion to its probability density in beta($n/2$, η) through the cumulative distribution and quantile functions of the beta distribution, with forward and backward proposal probabilities equal to the cumulative probability contained within each window centered on the forward state y and backward state y', i.e. found by integrating the beta probability density function between each set of bounds, which has positive density in (0,1) and zero density elsewhere. Resampling u', meanwhile, can be accomplished by sampling n - 1 normal random variables with mean = u and variance = tuning parameter v, and then rescaling this vector to unit length. By rescaling, we permit infinitely many samples from the aforementioned normal distributions to correspond to the same u', all falling on the vector stretching from the origin through u'. Forward and backward proposal probabilities can be obtained by integrating along this vector the multivariate normal probability distribution function with mean equal to the target state u or u' and covariance matrix diagonal with entries v. However, symmetry in the proposal distributions results in these integrals evaluating to the same value, and so the proposal ratio for this step can be set to 1.

As yet, the above proposal distribution samples only correlations of the final row and column of some correlation matrix R = $U^T U$. Its aggressiveness is controlled by two tuning parameters, w and v, though in practice we find that setting w to some small value, such as 0.1, and tuning only v is sufficient to achieve desired acceptance probabilities on moderately informative target distributions, such as those used in the second experiment below. For more informative target distributions, tuning w may also become necessary. Working on only the last dimension of U can be inefficiently accommodated by recomposing R, permuting, and refactoring. However, we can more cheaply remain in upper Cholesky form through two $O(n^2)$ steps, a rank-one update and downdate involving a series of Givens rotations, implemented in the R-package *mgcv* (**?**) as the function `choldrop`, followed by the solution of a triangular system of equations to reinsert the removed column in the final place, accomplished with base-R (**?**) function `backsolve`. To return the Cholesky factor to its original order, this procedure could be performed n - i times, where i is the index of the character deleted and reinserted. This is still more efficient than Cholesky factorization, but up to *2n* times more costly than the small burden of permutation index bookkeeping.

3 VALIDATION

Having implemented this proposal distribution in R, we seek to validate its performance in simulation, first by sampling uniformly from the space of correlation matrices by setting the acceptance probability to the ratio of proposal probabilities, implicitly sampling from an LKJ(1) with dimension 10. After having done so, we sample directly and independently from the analytical distribution, and visually inspect quantile-quantile plots of the marginal correlations and matrix determinants (Figure **??**).

Noting that these fall along the 1-to-1 line, we move on to explore sampling from a more tightly constrained distribution of correlation matrices, that defined by the joint posterior distribution of correlation matrices of a multivariate normal random variable with mean and variances known and equal to 1 and 10 x 10 correlation matrix drawn from an LKJ($\eta = 1$) (Figure **??**).

Specifically, this distribution represents the compromise between our flat LKJ($\eta = 1$) prior and the information contained in 50 samples from the above described multivariate normal, with target distribution density equal to the probability densities of our 50 draws with correlation matrix sampled and means and variances fixed. Comparing the samples obtained using the novel proposal distribution to those using the conventional sliding window, we first visually examining marginal trace plots and histograms and find that they closely overlap. Following that, we construct a similar figure as above, visualizing quantile-quantile plots and covariance patterns in both sets of samples and noting that they hew close to the 1-to-1 line, indicating that the same distribution is being sampled in both cases (Figure **??**).

Then, we coerce both arrays of correlation matrices to `mcmc.list` objects and evaluate Gelman and Rubin's Convergence Diagnostic (**?**) in the R package *coda* (**?**), finding for all pairwise correlations upper 95% confidence interval values of 1.00 and a multivariate $\hat{R}$ of 1. Effective sample sizes (ESS) of marginal correlations for both chains together ranged between 3,793 and 14,483, though much of those were from the chain using our novel proposal distribution, whose ESS ranged between 3,279 and 8,052, rather than from the sliding window chain, whose ESS ranged between 458 and 6,482, despite the latter having been tuned to have elementwise acceptance probabilities of 0.234 and being run for fivefold the number of iterations.

4 CONCLUSION

Thus, we have described, implemented, and validated a novel proposal distribution for correlation matrices. Our validation used a relatively small 10 x 10 correlation matrix — much greater improvements would be found with matrices of higher dimensionality. Often, Cholesky factorization counts among the most computationally steps of phylogenetic likelihood calculation in multivariate Brownian models, and circumventing this step by making proposals directly to the Cholesky form should enable tremendous gains to efficiency when performing inference of the character evolutionary processes governing the evolution of multiple traits. Even if one desires proposals directly on correlation matrices, making smarter proposals to multiple correlations simultaneously without the need for expensive PSD checks and rejection sampling should offer a substantial improvement to the use of more naive proposal distributions that do not intrinsically respect their constraints and properties.

5 FIGURES

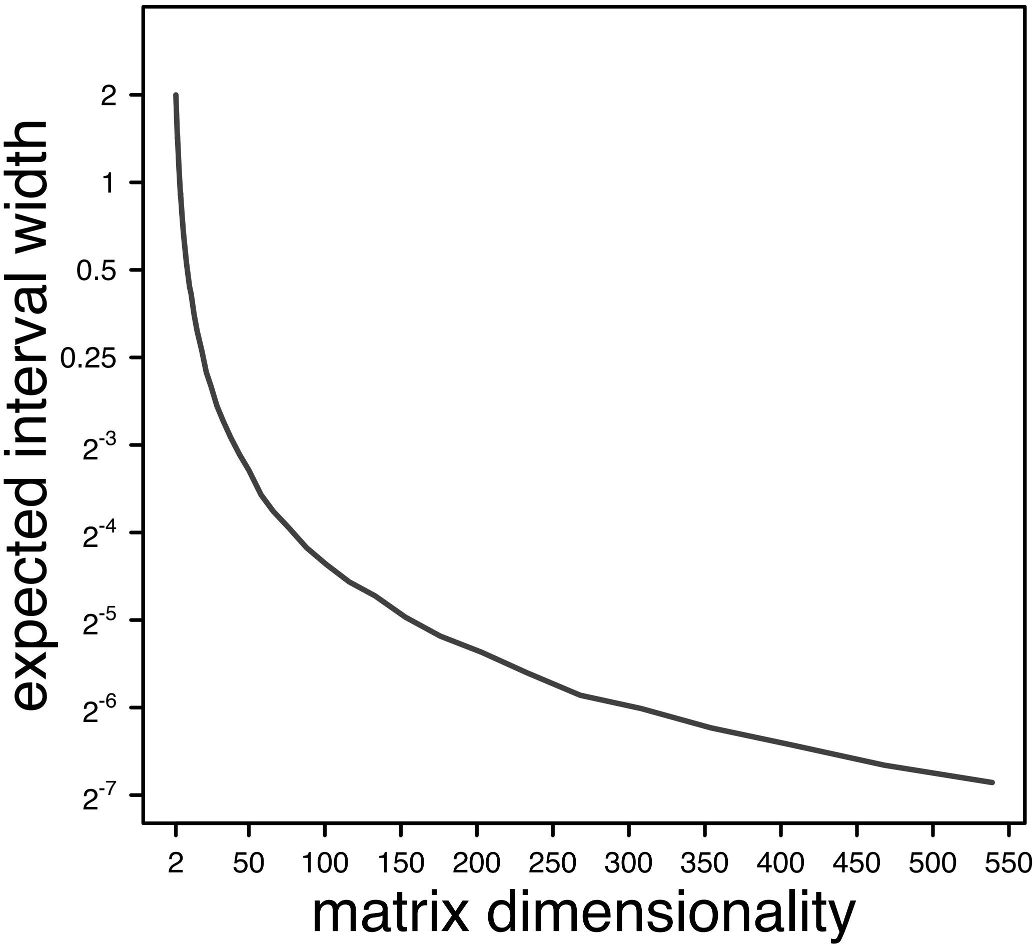

Figure 1: Depicting the relationship between how much space, in expectation, is available in a random correlation matrix of given dimensionality for valid proposals to marginal correlation coefficients. Quantities approximated using Monte Carlo simulation using samples from an LKJ($\eta = 1$). A random correlation coefficient was then chosen and perturbed in both directions by increasing amounts until the PSD constraint was violated.

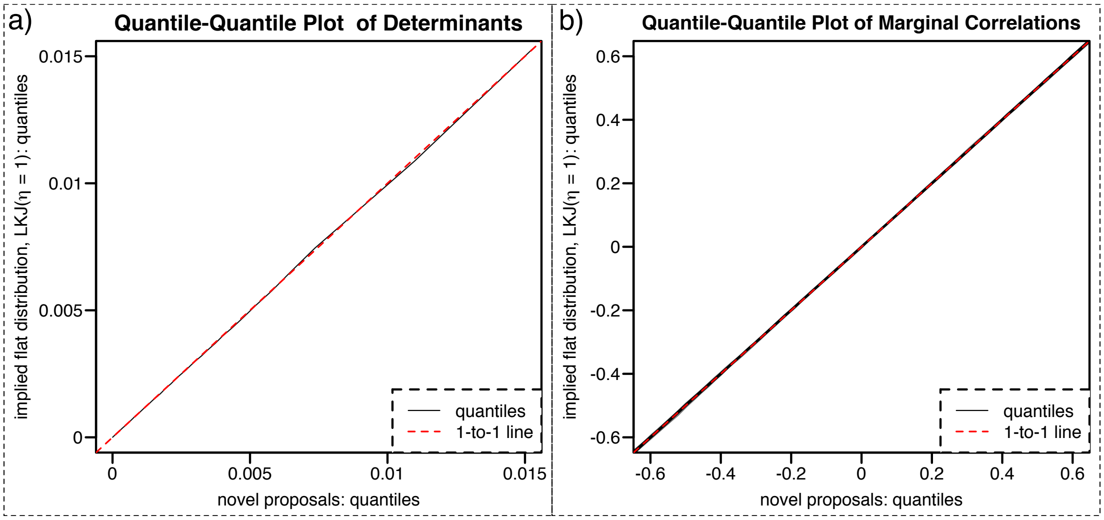

Figure 2: Depicting an initial experiment in the correct specification of the novel proposal distribution described herein. Target distribution density ratio was set to 1, implicitly sampling 10 x 10 correlation matrices from an LKJ($\eta = 1$). Samples were then generated from this distribution directly, and the quantiles of marginal correlation coefficients and matrix determinants compared.

1	-0.224	-0.175	-0.406	0.295	-0.028	0.199	-0.219	-0.211	-0.275
-0.224	1	0.365	0.326	0.008	-0.004	-0.026	0.235	0.016	0.2
-0.175	0.365	1	0.49	-0.139	0.539	-0.497	0.678	0.572	0.372
-0.406	0.326	0.49	1	-0.493	0.626	-0.134	0.324	0.621	0.22
0.295	0.008	-0.139	-0.493	1	0.009	0.041	0.044	-0.387	0.2
-0.028	-0.004	0.539	0.626	0.009	1	-0.246	0.393	0.451	0.363
0.199	-0.026	-0.497	-0.134	0.041	-0.246	1	-0.882	0.038	-0.417
-0.219	0.235	0.678	0.324	0.044	0.393	-0.882	1	0.233	0.523
-0.211	0.016	0.572	0.621	-0.387	0.451	0.038	0.233	1	-0.079
-0.275	0.2	0.372	0.22	0.2	0.363	-0.417	0.523	-0.079	1

Figure 3: The 10 x 10 covariance matrix used to generate samples from a multivariate normal distribution in the "informative target distribution" experiment. Generated by sampling from an LKJ($\eta = 1$) distribution after calling `set.seed(1)` in *R*.

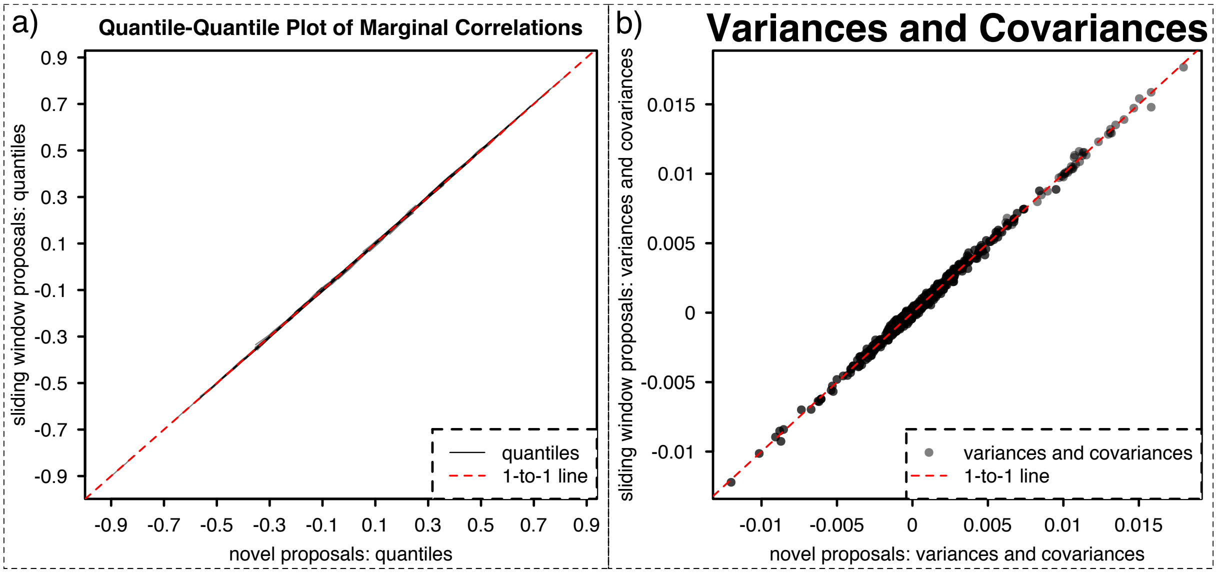

Figure 4: A visual comparison of MCMC output obtained using both our novel correlation matrix proposal distribution and a more standard uniform sliding window proposal distribution. In a), quantiles of each marginal correlation coefficient are plotted. In b), the variances and covariances of each marginal correlation coefficient for both chains are plotted.