# Appendix A: Citation patterns of Bayesian software packages

We analyzed the annual number of citations to the major Bayesian software packages across fields by accessing the ISI Web of Science Core Collection on 8 March 2016. Many citations were made to user manuals, books, and book chapters, and therefore citations were obtained from the Cited Reference Search. Citations to all years and versions of user manuals were included. Each citing paper was counted only once for each software package, even if it cited multiple papers describing a software package. We also excluded the year 2016 due to incomplete records.

We found *n*=5,234 citations, after filtering, in *n*=1,377 unique journals. The following references were included in the analysis:

**Stan**: [Hoffman and Gelman (2014](#_ENREF_6)); [Stan Development Team (2016](#_ENREF_23)); [Carpenter *et al.* (in press](#_ENREF_3))

**JAGS**: [Plummer (2003](#_ENREF_15)), [Plummer (2015](#_ENREF_16))

**BUGS**: [Spiegelhalter *et al.* (1996](#_ENREF_21))

**WinBUGS**: [Lunn *et al.* (2000](#_ENREF_11)); [Spiegelhalter *et al.* (2003](#_ENREF_19)); [Sturtz, Ligges and Gelman (2005](#_ENREF_24)); [Thompson, Palmer and Moreno (2006](#_ENREF_26)); [Lunn *et al.* (2012](#_ENREF_9))

**OpenBUGS**: [Thomas (2004](#_ENREF_25)); [Lunn *et al.* (2009](#_ENREF_10) ); [Spiegelhalter *et al.* (2014](#_ENREF_20))

# Appendix B: Model files and code

This study can be reproduced using free, open source software. The interested reader can find model files, simulation code, and implementation of the static HMC and NUTS algorithms, written in R, online at: <https://github.com/colemonnahan/gradmcmc>.

# Appendix C: Measuring MCMC efficiency

There are many factors contributing to which software package is ‘best’ for a real problem. Aspects such as learning curve, model development time, diagnostics tools, and documentation typically play a large role. Here, we assume that run time is a limiting factor preventing inference for an applied problem, so that more ‘efficient’ algorithms are desired. There are two key aspects of a metric that measures efficiency: run time and the minimum effective number of samples (ESS) produced.

Run time has been quantified using both log density evaluations and processor time ([e.g., Thompson 2010](#_ENREF_27)), with the former being preferable when comparisons are within the same software. However, here we are comparing across software platforms, and further, Stan is also calculating gradients so model evaluations are not directly comparable. As such, we adopt processor time as the metric of run time for this study. Run time includes the warmup period but not the time to compile Stan models.

ESS is typically lower than the nominal sample size because samples drawn using MCMC are dependent (i.e., autocorrelated) and thus contain less information than the nominal sample size. ESS must be approximated from the samples ([e.g., Thompson 2010](#_ENREF_27)), and to date have usually been calculated using a package like CODA ([Plummer *et al.* 2006](#_ENREF_17)). However, a new formula presented in [Hoffman and Gelman (2014](#_ENREF_6)) is more reliable and accurate, and thus generally preferred. Although the formula is the default in Stan it is available for any MCMC output (e.g., with JAGS) by using the function monitor in the R package rstan ([Stan Development Team 2015](#_ENREF_22)), which we use to calculate ESS for all models in this study. Since each parameter has a different ESS, we take the minimum across parameters as the bottleneck of information, following the lead of other studies ([Girolami & Calderhead 2011](#_ENREF_4); [Hoffman & Gelman 2014](#_ENREF_6)).

Thus *efficiency* has a specific definition throughout this study: the estimated number of effective samples drawn per unit time for the slowest mixing parameter.

# Appendix D: Efficient parameterizations for hierarchical models

Hierarchical models can be particularly challenging for MCMC samplers due to the geometry of the hypervariance and random effects. For small values of the hypervariance, each random effect will be constrained to be small. As the hypervariance increases, so will the distribution of the random effects, creating a funnel shaped relationship between the two (see Fig. 3 of [Betancourt and Girolami (2015](#_ENREF_1))). The more exaggerated this geometric shape, the harder it is generally for MCMC samplers to efficiently sample from the posterior ([Papaspiliopoulos, Roberts & Skold 2007](#_ENREF_14)).

This is a problem for HMC in particular because the optimal step size will vary with the hypervariance: smaller step sizes are appropriate for smaller hypervariances and vice versa. Further, this shape cannot be corrected by a global rotation or scaling (via the mass matrix), and the slow mixing subsequently impacts the tuning of the mass matrix because the sampler will not explore the whole space during warmup ([Betancourt & Girolami 2015](#_ENREF_1)).

Fortunately this issue can often be mitigated, at least to some degree, by reparameterizing the hyperdistribution from a *centered* into a *noncentered* form ([Papaspiliopoulos, Roberts & Skold 2007](#_ENREF_14)). This reparameterization technique may be unfamiliar to ecologists, but can make substantial improvements to efficiencies for both HMC and other samplers as well ([Betancourt & Girolami 2015](#_ENREF_1)). Noncentering works on all distributions in the location and scale family, as detailed in [section 20.1 of Stan Development Team (2016](#_ENREF_23)), but here we demonstrate this technique on normally distributed random effects. Let µ and σ be the hyperparameters, then let



The random variable *X1* is the centered form and the natural way to parameterize the model, while *X2* the noncentered parameterization. Here both *X1* and *X2* have the same distribution but the parameters (and hence the posterior geometries) are different. In the centered form we directly model the random effects, *X1.* In the noncentered form the parameters are *Z*, and the random effects used in the model, *X2*, are derived from these.

The benefit of this is that the correlation between σ2 (the hypervariance) and the random effects is minimized, mitigating the funnel shaped relationship between them. However, this increases the correlation between the fixed effects in the model. Further, if the data are informative about σ2 (i.e., it has a narrow marginal posterior distribution) then the funnel is effectively truncated and the noncentered form may actually introduce more difficult correlations for the samplers. Thus, which parameterization is most efficient will depend on the situation: model structure, data, and priors ([Papaspiliopoulos, Roberts & Skold 2007](#_ENREF_14)).

Fortunately, it is typically straightforward in practice to switch between parameterizations for both Stan and JAGS, thus testing both forms as part of model development is straightforward and an approach we advocate. If the funnel shape cannot be eliminated or mitigated, HMC may have a hard time exploring all parts of the posterior. In this case, it is recommended to run a chain with smaller step sizes (i.e., set adapt\_delta closer to 1) and verify it explores the same posterior space ([Betancourt & Girolami 2015](#_ENREF_1)).

# Appendix E: Case study details

Here we provide further details of our case studies, summarized in Table 1. Note that model files for both Stan and JAGS are available at <https://github.com/colemonnahan/gradmcmc/tree/master/models>. For hierarchical models, we fit both a centered and noncentered form for the random effects, with the latter having names appended with ‘\_nc’ to distinguish them.

## MVND and MVNC

Our first simulation model is a multivariate normal. MCMC methods are not needed here, but this model provides a convenient testing framework because it is easy to scale dimensionality and complexity (via the covariance structure). We use this model in two ways. First, we generate a 50 by 50 covariance matrix with a mixture of weak and strong correlations using an inverse Wishart distribution (with 50 df and diagonal covariance), as done in [Hoffman and Gelman (2014](#_ENREF_6)), varying the dimensionality by taking subsets of this matrix (MVND) from 2 to 50. For each subset we also run a diagonal covariance matrix (i.e., independent) to differentiate the effect of model size vs. complexity (correlations).

Second, we generate a covariance matrix with constant correlations () for all off-diagonal entries. We varied ρ from 0 (independent) to 0.95, and for each level of ρ we run the model for 2, 25, and 50 dimensions, again to differentiate effects of size and correlation. We name this model MVNC.

Samplers which update a single parameter at a time (e.g., Gibbs and slice sampling, typically) are known to be sensitive to correlations or ‘scale invariant.’ Conversely, HMC, which updates all parameters simultaneously, is known to be sensitive to variable scales but not correlations, and is thus ‘rotation invariant.’ However, if the mass matrix closely matches the global covariance structure (and it is constant across the surface) then HMC will be scaled appropriately ([see p.22 of Neal (2011](#_ENREF_13))). These differences motivate our decision to test both MVND, which has different scales, and MVNC which has the same scale but stronger correlations.

For both these models we used broad, non-conjugate uniform priors, so that Gibbs sampling was not possible and note that slice sampling was utilized by JAGS. These models mimic fixed effects models with non-conjugate priors.

## Growth

The second simulation model, ‘Growth,’ is a non-linear somatic growth model with repeated measures ([Schnute 1981](#_ENREF_18)). This model has two sets of random effects (two measurements for each individual), and fixed effects for hyperparameters and a term controlling the shape of the growth pattern.

We simulated data by drawing from independent hyperdistributions for *k* (growth rate) and (mean maximum length), both on the log scale, sampling at 5 random ages, and then adding normal measurement error to length on the log scale. This simulates resampling the same individual multiple times over its lifetime, as would be done for example in a controlled lab study. Our estimation model matched the structure of the simulated data, so that each animal had a corresponding random effect for log*k* and log*L*max, as well as fixed effects for hypermeans, hypervariances, an observation variance term for the normal likelihood.

Similar to the multivariate normal examples, this model let us explore the effect of dimensionality (by increasing the number of individuals), and how that interacted with noncentering the random effects. We ran the model with number of individuals from 2 to 316. We used an adapt\_delta of 0.8 for both the centered and noncentered forms, although higher values would be necessary to minimize divergent transitions for models with few individuals in practice. In this case, we favored a constant target acceptance rate to more easily compare performance and test for divergence.

The estimation model is:



Priors were specified as:



Simulated values were



## Redkite

This model is described in detail in section 8.4 of ([Kéry & Schaub 2012](#_ENREF_7)). The data come from marking 1480 nestlings and 152 adults over the previous 50 years. The model estimates age-specific survival probabilities of three age classes: juveniles, subadults and adults. This model contains an informative prior on juvenile survival, because it is often difficult to estimate with these kinds of data.

This model was converted to Stan and posted online at <https://github.com/stan-dev/example-models/tree/master/BPA> (last accessed 5/8/2016), while the original BUGS code was taken from <http://www.vogelwarte.ch/de/projekte/publikationen/bpa/complete-code-and-data-files-of-the-book.html> (last accessed 5/8/2016).

## Swallows

The ‘Swallows’ model estimates survival of birds accounting for survival and detection using a state-space construction, while linking survival and detection probabilities to environmental covariates. It also includes a complex hierarchical design, with year and family random effects for survival, and family random effects for detection ([Gruebler & Naef-Daenzer 2008](#_ENREF_5); [Korner-Nievergelt *et al.* 2015; Section 14.5](#_ENREF_8)).

The original model code can be found online at <http://www.oikostat.ch/blmeco/ch14/CJS_swallows.stan> (last accessed 5/4/2016). We modified this version to include recommended priors for the hyper-variance terms, additional data preprocessing, and a few computational improvements (such as vectorization). [Korner-Nievergelt *et al.* (2015](#_ENREF_8)) found that their BUGS implementation was prohibitively slow compared to Stan. However, that implementation tracked discrete states for each animal in each time step. That is, it did not marginalize out the discrete parameters. When we recreated the model in JAGS we used the same parameterization as Stan. Thus it seems likely that the major increase in efficiency that we found was due to the parameterization, not the actual software platform or algorithm.

## Logistic

This model is a state-space population dynamics model with a logistic population growth function applied to South Atlantic tuna catch and effort data. Process errors, observation variance, and biological parameters are all estimated, some of which have informative priors derived from meta-analyses. This model is described in detail in ([Millar & Meyer 2000](#_ENREF_12)), including the original BUGS code. We used adapt\_delta of 0.95 for both forms to minimize divergences.

## Wildflower

This model was analyzed in [Bolker *et al.* (2013](#_ENREF_2)), and looks at individual flowering success of Bitterroot milkvetch (*Astragulus scaphoides*) as a function of the last year’s stage and seed production. It is a binomial generalized linear mixed effects model, with three sets of random effects, two of which are crossed. The first set acts as an intercept on year, while the latter two are individual effects for slope and intercept. The fixed effects act as intercept on stage, a global slope, and hypervariances.

Following the original authors, we did not center the predictor variables. However, we did introduce the recommended half-Cauchy priors for the variance terms, instead of bounding their values and using a uniform prior as originally done.

# References

Betancourt, M. & Girolami, M. (2015) Hamiltonian Monte Carlo for hierarchical models. *Current Trends in Bayesian Methodology with Applications***,** 79.

Bolker, B.M., Gardner, B., Maunder, M., Berg, C.W., Brooks, M., Comita, L., Crone, E., Cubaynes, S., Davies, T., de Valpine, P., Ford, J., Gimenez, O., Kery, M., Kim, E.J., Lennert-Cody, C., Magnusson, A., Martell, S., Nash, J., Nielsen, A., Regetz, J., Skaug, H. & Zipkin, E. (2013) Strategies for fitting nonlinear ecological models in R, AD Model Builder, and BUGS. *Methods in Ecology and Evolution,* **4,** 501-512.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M.A., Guo, J., Li, P. & Riddell, A. (in press) Stan: a probabilistic programming language. *Journal of Statistical Software*.

Girolami, M. & Calderhead, B. (2011) Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B-Statistical Methodology,* **73,** 123-214.

Gruebler, M.U. & Naef-Daenzer, B. (2008) Fitness consequences of pre- and post-fledging timing decisions in a double-brooded passerine. *Ecology,* **89,** 2736-2745.

Hoffman, M.D. & Gelman, A. (2014) The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research,* **15,** 1593-1623.

Kéry, M. & Schaub, M. (2012) *Bayesian population analysis using WinBUGS: a hierarchical perspective*. Academic Press.

Korner-Nievergelt, F., Roth, T., von Felten, S., Guélat, J., Almasi, B. & Korner-Nievergelt, P. (2015) *Bayesian data analysis in ecology using linear models with R, BUGS, and Stan: including comparisons to frequentist statistics*. Academic Press.

Lunn, D., Jackson, C., Best, N., Thomas, A. & Spiegelhalter, D. (2012) *The BUGS book: A practical introduction to Bayesian analysis*. CRC press.

Lunn, D., Spiegelhalter, D., Thomas, A. & Best, N. (2009) The BUGS project: Evolution, critique and future directions. *Statistics in medicine,* **28,** 3049-3067.

Lunn, D.J., Thomas, A., Best, N. & Spiegelhalter, D. (2000) WinBUGS-a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing,* **10,** 325-337.

Millar, R.B. & Meyer, R. (2000) Non-linear state space modelling of fisheries biomass dynamics by using Metropolis-Hastings within-Gibbs sampling. *Journal of the Royal Statistical Society Series C-Applied Statistics,* **49,** 327-342.

Neal, R.M. (2011) MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo,* **2**.

Papaspiliopoulos, O., Roberts, G.O. & Skold, M. (2007) A general framework for the parametrization of hierarchical models. *Statistical Science,* **22,** 59-73.

Plummer, M. (2003) JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. *Proceedings of the 3rd international workshop on distributed statistical computing*, pp. 125.Technische Universit at Wien, Wien, Austria.

Plummer, M. (2015) JAGS Version 4.0.0 user manual. .

Plummer, M., Best, N., Cowles, K. & Vines, K. (2006) CODA: Convergence Diagnostics and Out Analysis for MCMC. *R News,* **6,** 7-11.

Schnute, J. (1981) A versatile growth model with statistically stable parameters. *Canadian Journal of Fisheries and Aquatic Sciences,* **38,** 1128-1140.

Spiegelhalter, D., Thomas, A., Best, N. & Lunn, D. (2003) WinBUGS user manual.

Spiegelhalter, D., Thomas, A., Best, N. & Lunn, D. (2014) OpenBUGS user manual, version 3.2.3. MRC Biostatistics Unit, Cambridge.

Spiegelhalter, D.J., Thomas, A., Best, N., Gilks, W. & Lunn, D. (1996) BUGS: Bayesian inference using Gibbs sampling. *Version 0.5ii,* **19**.

Stan Development Team (2015) RStan: The R interface to Stan. Version 2.9.0. <http://mc-stan.org/rstan.html>.

Stan Development Team (2016) Stan modeling language users guide and reference manual, version 2.11.0.

Sturtz, S., Ligges, U. & Gelman, A.E. (2005) R2WinBUGS: a package for running WinBUGS from R. *Journal of Statistical Software,* **12,** 1-16.

Thomas, A. (2004) OpenBUGS. URL <http://mathstat>.

Thompson, J., Palmer, T. & Moreno, S. (2006) Bayesian analysis in Stata using WinBUGS. *The Stata Journal,* **6,** 530-549.

Thompson, M.B. (2010) A comparison of methods for computing autocorrelation time. *arXiv preprint arXiv:1011.0175*.