How to fit Bayesian models and influence people *or* How to do Bayesian model fitting in ecology *or* The best way to be a Bayesian in ecology today

EM Wolkovich and WD Pearse and M Betancourt? and TJ Davies?

August 11, 2023

Abstract

Improvements in algorithms and computational speed have heralded a new a era of Bayesian model fitting. Models are easier to fit, faster to run, and more flexible to fit, making them ideal for many of the complexities of ecological sampling designs. This has opened up Bayesian approaches to a new world of users, but how best to start using and implementing Bayesian approaches in ecology is still somewhere between old rules and methods and the new school where Bayesian approaches also include a specific workflow of not just how to do stats, but how to do science. Here we review a simple form of a Bayesian workflow that integrates mechanistic and statistical models with a computational toolkit that we argue could accelerate ecological science. [Maybe along the way we highlight common practices in ecology that are stupid as all tomorrow—as our approach makes clear—and outline best practices for jumping into the wonderful surf of the happy Bayesian ocean.]

1 Main text

Recent years have seen the explosion of Bayesian models across fields (CITES), including ecology. This change comes in part from increased computational power, but more from new algorithms (e.g., Hamiltonian Monte Carlo) that have made models faster and arguably easier to fit. Capitalizing on these advances R packages such as brms and rstanarm have seen much wider use than the previous generation of packages that streamlined fitting a set of pre-determined models using Bayesian approaches (MCMCglmm. Bayesian approaches, long heralded as more powerful, flexible and especially adept at capturing the multiple levels of variance in ecological data, seem poised to help ecology as a discipline advance in leaps and bounds.

Ecology itself seems poised to leap out of the jungle and land into the role society is asking of it as growing global challenges demand more predictive models of how communities and ecosystems will shift with climate change, habitat degradation and all that jazz. Ecology is now three decades on after the start of the synthesis movement, which required more advanced statistical and computational approaches of ecology to test fundamental theory across systems (CITES). Early concerns have given way to a widespread appreciation of the value of these new approaches, alongside a resounding fervor over 'natural history.' At the same time, the growing anthropogenic challenges have increased the need for predictive models and forecasts from ecologists.

The result is that the average ecologist today has a diversified skillset compared to that

of decades ago. Ecology has long been a field with deep statistical training, but in many ways the modern ecologist is now also expected to be computational—able to handle large datasets, produce repeatable workflows and help translate models into forecasts for planning. Many ecologists now bridge field and computational methods. As such, the rise of Bayesian approaches is especially timely for ecology.

Bayesian approaches are in no way new to ecology, as they have long been used in certain subdisciplines related to estimating population sizes of things people want to eat or manage. For example, wildlife biology widely uses mark-recapture data and associated models to estimate population sizes; these methods almost always use Hidden Markov models (HMM), which can rarely be fit without Bayesian methods (CITES). Similarly, fisheries biology has tended towards more complex models—such as state-space models—to estimate fish stocks, which are similarly easiest to fit with Bayesian approaches (CITES). For decades, Bayesian training in ecology has focused on these aims, but as data and methods change, so has how ecologists are using Bayesian models.

Today, as large-scale ecological data becomes available for more diverse systems and for questions addressing other aims, more ecologists are using Bayesian models in new ways. Yet, at the same time, training in statistics and in Bayesian modeling in particular may struggle to keep pace. Bayesian approaches provide a pathway to powerful models that can transform how we understand our systems, but they can also lead to pitfalls most ecologists are not trained to notice or deal with. These pitfalls can be avoided by approaching Bayesian analyses through specific workflows (CITES), which themselves are built on a process of how to do not just stats, but science. Here we describe a broadly generalizable workflow for Bayesian analysis and show how it can revolutionize training in ecology by integrating more model building and model understanding. Once you start doing this workflow, your scientific life will never be the same.

1.1 Box by WPD with 3 paragraph explanation of Bayes

1.2 A brief overview of the benefits—and pitfalls—of Bayesian models

1. You can fit any model you want! This is so great as it blurs the line between conceptual models, mechanistic models, process-based models, mathematical models (theory) and statistical models. Yay!
2. Because you can fit whatever you want, you can get numbers that really make sense – numbers you may really want to estimate. This focus on effect sizes and what you want to estimate, moves away from p-values. Suddenly you can stop saying 'effect size' as some unitless Guretevitch value and say 'change in days per degree of warming under X model or such.'
3. You can fit any model you want! So you can fit stuff that may be almost entirely uninfluenced by your data if you have no idea what you're doing. Oh dear.
4. So, how you do get all the benefits of Bayesian without doing something stupid?

1.3 Our simple Bayesian workflow

Below I outline a basic Bayesian workflow for someone using `Stan`. This is a short overview of a much deeper topic. See Bayesian statistics and modelling for a little more depth, including how to develop priors and your basic model formulation.

Many steps should be familiar to statistical ecologists, but are often overlooked. We provide further steps particular to Bayesian (prior predictive checks).

Note that you might need to start simple and build up to get all these steps to work, but this is my recommended approach to the basic workflow.

1. Check your code. Write down your model – I recommend as basic math, then write it in `Stan` and simulate one set of test data (often in `R`). I recommend you do this first because it will flesh out any issues in your simulated data (`R`) code, which you need right away (but you don't need the `Stan` code until step 3).
   (a) Write your `Stan` code.
   (b) In `R`, write simulated data where you **write out all your parameters**, write x and generate y from your model. So, if I am doing simple regression ($y = mx + b$, with error ϵ) then I would have to assign values to m, b and ϵ and I would generate a vector of x values, then I would simulate y from those values.
   (c) Run your `Stan` code on your simulated data. Check that your `Stan` code returns your model parameters, if not, check your code, set your error lower and/or sample size higher. Keep checking until your `Stan` output matches your parameters (note check your `Stan` code and your simulated data code) and you trust both.
2. Prior predictive checks: Check yo' priors.
   (a) Take your aforementioned `R` code and set up priors for each parameter (e.g., distributions for m, b, ϵ). **In contrast to above, where you just set each parameter to one value, here you you want to draw multiple values for each parameter and visually check the output.**
   (b) How do I check the output? Just like posterior predictive checks, this is up to you! At a minimum I recommend thinking of plots you will make with your model in the end (for publications) and plotting that given different prior values.
   (c) Your goal here is to check that your priors are reasonable, if they are not, adjust them.
   (d) Unlike in Step 1, you do **not** need to run `Stan`—you have your parameters so you can just simulate data from them, and then plot, examine etc.. No `Stan` at this step.
3. Now you can run your model on your real data!
   (a) Check the output, if you have divergent transitions, you need to re-parameterize your model (may tried a non-centered model or such).
   (b) I recommend ShinyStan here.
4. Posterior predictive checks: How good or bad does your model do compared to your data?
   (a) Grab the parameter values from your fitted `Stan` model. In posterior predictive checks *these are the parameter values you use to simulate new data.*
   (b) You can adapt your `R` code for simulating data above, but use your estimated parameters from your fitted model.
   (c) What should I look at? Ah, just like in prior predictive checks, you need to decide. Classic things are to look at the mean of 100 or so simulations of new data you generate versus your **real** data. Also try the SD. Plot things! Look at the distribution. If you use the generated quantities block in `Stan` ShinyStan will automatically generate a few plots but think hard about more.
   (d) When does this get hard? In hierarchical models (and other models with hyperparameters) as you have multiple levels you can generate—you can use *your estimated parameters from your fitted model at all levels or generate your lower-level parameters* (for example, you can generate species means using your species μ and σ). You have to think about what you want your model to predict.
   (e) See how it is—do you see obvious problems caused by the distribution you selected or a grouping factor you're missing? If so, add it. And go back to Step 1.
   (f) (No `Stan` at this step.)

Benefits of this workflow

1. More fully integrates the mathematical model to statistical part of Bayesian – you have to write the model with test data before you fit to your data
2. Both the test data writing and the posterior predictive checks dive you deep into understanding your mechanistic-statistical model and that, in our experience, gives you WAY more insights and ideas into your biological model and—wait for it—your biological system!

1.4 Surprising things that happened to us that may happen to you if you follow this workflow

1. We got deeply in touch with the term **nonidentifiability** and how it can happen (model nonidentifiability and data + model nonidentifiability) ... now that you can fit any model you want, you see this happen (before, a Hessian problem may have stood in your way sometimes for these models, but also sometimes for perfectly good models)
2. Simplify! Simplify! Simplify! Once you do the workflow, you may end up like us: fitting fewer levels in your mixed effects models, fitting fewer interactions.
3. Don't cram the world into your model ... in contrast, value of the posterior for manipulation.
4. Plotting: The whole big wide world between plotting your raw data and plotting you model ... Plotting 'partials' of your model (remove the site effects, for example).

1.5 Challenges of the workflow compared to how we traditionally train ecologists (or, how this will reshape ecological training)

1. Good stats workflows bleed into what we expect of theoretical ecologists (and yet we act like non-theory folks should be trained differently).
2. Theory = simpler models = outcome of a good Bayesian workflow (often)

Moving away from old school Bayesian and into the light – Read this section to the tune of 'Let it go' from *Frozen*

1. Everyone can be a Bayesian, not just wildlife and fisheries biologists (aka HMM and state-space people)
2. Please, stop going on and on about priors.
3. Conjugate priors as the crystal deodorant of priors (check Dan Simpson quote)
4. Let go of 'random' versus fixed effects ideas
5. Let go of p-values and embrace numbers with units! ('I am arbitrary but my story is often told ... '

1.6 Conclusions

1. Ecologists cannot simulate their stats (or simple systems for that matter). Evolutionary biologists can. (And the field is better for it.)
2. Maybe hint at that you need these skills (and unit testing) given rise of AI?

Take home messages (maybe)

1. You should not fit a model you cannot simulate
2. Fit simpler models
3. Know your nonidentifiability

2 Outline

1. Intro
   (a) The explosion of Bayesian approaches recently – boom!
   (b) At the same time ecology has and is shifting (NCEAS etc.)
      i. 'The modern ecologist is computational.'
      ii. Intro should include working across skillsets in different fields of ecology.
   (c) Old Bayesian
   (d) So, where does that leave us?
   (e) Here we describe a broadly generalizable workflow for Bayesian analysis and show how it can revolutionize training in ecology by integrating more model building and model understanding.
   (f) Will writes 3 paragraph explanation of Bayes (this goes in a box)
2. General benefits—and pitfalls—of Bayesian
   (a) You can fit any model you want! This is so great as it blurs the line between conceptual models, mechanistic models, process-based models, mathematical models (theory) and statistical models. Yay!
   (b) Because you can fit whatever you want, you can get numbers that really make sense – numbers you may really want to estimate. This focus on effect sizes and what you want to estimate, moves away from p-values. Suddenly you can stop saying 'effect size' as some unitless Guretevitch value and say 'change in days per degree of warming under X model or such.'
   (c) You can fit any model you want! So you can fit stuff that may be almost entirely uninfluenced by your data if you have no idea what you're doing. Oh dear.
   (d) So, how you do get all the benefits of Bayesian without doing something stupid?
3. Introducing the very basic workflow (just explain it Lizzie!) ... end or somewhere add: Many steps should be familiar to statistical ecologists, but are often overlooked. We provide further steps particular to Bayesian (prior predictive checks).
4. Benefits of this workflow
   (a) More fully integrates the mathematical model to statistical part of Bayesian – you have to write the model with test data before you fit to your data
   (b) Both the test data writing and the posterior predictive checks dive you deep into understanding your mechanistic-statistical model and that, in our experience, gives you WAY more insights and ideas into your biological model and—wait for it—your biological system!
5. Surprising things that happened to us that may happen to you if you follow this workflow
   (a) We got deeply in touch with the term **nonidentifiability** and how it can happen (model nonidentifiability and data + model nonidentifiability) ... now that you can fit any model you want, you see this happen (before, a Hessian problem may have stood in your way sometimes for these models, but also sometimes for perfectly good models)
   (b) Simplify! Simplify! Simplify! Once you do the workflow, you may end up like us: fitting fewer levels in your mixed effects models, fitting fewer interactions.
   (c) Don't cram the world into your model ... in contrast, value of the posterior for manipulation.
   (d) Plotting: The whole big wide world between plotting your raw data and plotting you model ... Plotting 'partials' of your model (remove the site effects, for example).
6. Challenges of the workflow compared to how we traditionally train ecologists (or, how this will reshape ecological training)

   (a) Good stats workflows bleed into what we expect of theoretical ecologists (and yet we act like non-theory folks should be trained differently).
   (b) Theory = simpler models = outcome of a good Bayesian workflow (often)
7. Moving away from old school Bayesian and into the light – Read this section to the tune of 'Let it go' from *Frozen*
   (a) Everyone can be a Bayesian, not just wildlife and fisheries biologists (aka HMM and state-space people)
   (b) Please, stop going on and on about priors.
   (c) Conjugate priors as the crystal deodorant of priors (check Dan Simpson quote)
   (d) Let go of 'random' versus fixed effects ideas
   (e) Let go of p-values and embrace numbers with units! ('I am arbitrary but my story is often told ... '
8. Conclusions
   (a) Ecologists cannot simulate their stats (or simple systems for that matter). Evolutionary biologists can. (And the field is better for it.)
   (b) Maybe hint at that you need these skills (and unit testing) given rise of AI?

   Take home messages (maybe)
   (a) You should not fit a model you cannot simulate
   (b) Fit simpler models
   (c) Know your nonidentifiability

Stuff missing a home in the outline ....

1. 'Best practices' workflow (or just 'best practices'?)
2. Process-based models versus theory versus workflow
3. α and β power (Will writes this)
4. Community uncertainty and propagating uncertainty (Dietze)
5. Theoretical ecology is a more separated field in ecology compared to in evolution.

Things to decide/do

1. Do we need an example? Cherries or the project with Heather?
2. Sample code (could get Will to do this) including ...
   (a) Bad interaction
   (b) Bad prior
   (c) Including study and species (non-identifiability)

3 Refs to check out...

Very short keynote talk I gave.

Workflow in van der schoot (the paper with Ruth King?)
Gelman has a workflow paper

What Betancourt has.
Gabry has visualization in the workslow paper.

4 My old tutorial!

One Bayesian Workflow:

Below I outline a basic Bayesian workflow for someone using `Stan`. This is a short overview of a much deeper topic. See Bayesian statistics and modelling for a little more depth, including how to develop priors and your basic model formulation.

Note that you might need to start simple and build up to get all these steps to work, but this is my recommended approach to the basic workflow.

1. Check your code. Write down your model – I recommend as basic math, then write it in `Stan` and simulate one set of test data (often in `R`). I recommend you do this first because it will flesh out any issues in your simulated data (`R`) code, which you need right away (but you don't need the `Stan` code until step 3).
   (a) Write your `Stan` code.
   (b) In `R`, write simulated data where you **write out all your parameters**, write x and generate y from your model. So, if I am doing simple regression ($y = mx + b$, with error ϵ) then I would have to assign values to m, b and ϵ and I would generate a vector of x values, then I would simulate y from those values.
   (c) Run your `Stan` code on your simulated data. Check that your `Stan` code returns your model parameters, if not, check your code, set your error lower and/or sample size higher. Keep checking until your `Stan` output matches your parameters (note check your `Stan` code and your simulated data code) and you trust both.
2. Prior predictive checks: Check yo' priors.
   (a) Take your aforementioned `R` code and set up priors for each parameter (e.g., distributions for m, b, ϵ). **In contrast to above, where you just set each parameter to one value, here you you want to draw multiple values for each parameter and visually check the output.**
   (b) How do I check the output? Just like posterior predictive checks, this is up to you! At a minimum I recommend thinking of plots you will make with your model in the end (for publications) and plotting that given different prior values.
   (c) Your goal here is to check that your priors are reasonable, if they are not, adjust them.
   (d) Unlike in Step 1, you do **not** need to run `Stan`—you have your parameters so you can just simulate data from them, and then plot, examine etc.. No `Stan` at this step.
3. Now you can run your model on your real data!
   (a) Check the output, if you have divergent transitions, you need to re-parameterize your model (may tried a non-centered model or such).
   (b) I recommend ShinyStan here.
4. Posterior predictive checks: How good or bad does your model do compared to your data?
   (a) Grab the parameter values from your fitted `Stan` model. In posterior predictive checks *these are the parameter values you use to simulate new data.*
   (b) You can adapt your `R` code for simulating data above, but use your estimated parameters from your fitted model.
   (c) What should I look at? Ah, just like in prior predictive checks, you need to decide. Classic things are to look at the mean of 100 or so simulations of new data you generate versus your **real** data. Also try the SD. Plot things! Look at the distribution. If you use the generated quantities block in `Stan` ShinyStan will automatically generate a few plots but think hard about more.

(d) When does this get hard? In hierarchical models (and other models with hyperparameters) as you have multiple levels you can generate—you can use *your estimated parameters from your fitted model at all levels or generate your lower-level parameters* (for example, you can generate species means using your species μ and σ). You have to think about what you want your model to predict.

(e) See how it is—do you see obvious problems caused by the distribution you selected or a grouping factor you're missing? If so, add it. And go back to Step 1.

(f) (No `Stan` at this step.)