Hi Jim,

A good number of years ago we did an assessment of the NEON plant diversity data to assess assumptions of the initial design. One component was sensitivity analysis - determining whether a 20% year-to-year change in species abundance could be detected and how the number of plots sampled affected that detectability. Now, with more years of data, I’m attempting to recreate the analysis. Instead of modeling each species separately with linear mixed models, I thought to use GJAM. My plan was to:

1. **Fit a model for each site**  
   Use GJAM to model percent cover of plant species at each NEON site, with predictors limited to year (as a factor) and land cover type (from nlcdClass).
2. **Generate posterior predictions**  
   Generate posterior predictions for species abundance under observed covariate conditions to establish a baseline distribution.
3. **Simulate a 20% increase in species abundance**  
   Apply a 20% increase in species abundance to test the model’s capacity to detect ecologically meaningful changes.
4. **Evaluate sensitivity to sample size**  
   Repeat the modeling and prediction steps using reduced subsets of plots within each site to assess how detection probabilities vary with sample size.
5. **Compare predictions and assess detectability**  
   Compare posterior predictions under baseline and simulated-change scenarios to determine whether the model can reliably detect the imposed change in abundance, and how that ability varies with sample size.

We’re having a hard time figuring out how to generate baseline predictions using full posterior distributions (step 2) such that the model – including uncertainty and species covariance structures - can be applied to the abundance-modified data (step4) and used to compare posterior distributions (step 5). Attempts with gjamPredict() seem to result errors even when REDUCT = FALSE. As an alternative, we’ve tried writing a function that draws from the saved posteriors (betaBeta and sigmaSave) but run into problems related to how residual covariances are stored - errors with matrix dimensions and reshaping of posterior chains.

We considered a simpler approach: increasing the model’s predicted means by 20% to mimic the effect of higher abundance, without re-running the full posterior prediction. This might give some sense of effect size, but would not reflect uncertainty nor the model’s capacity to detect change across different sample sizes.  
  
Might you or one of your students be able to provide some direction? Is the application of GJAM appropriate to this kind of sensitivity analysis? Is the general logic — of reusing the model to evaluate known abundance changes by applying posterior uncertainty and species correlation — sound/possible/appropriate within the framework of GJAM?

Thanks so much. I’m new to much of this and doing a bit of doggy paddling in the deep end, I suppose that is appropriate for the summer season. I hope yours is great!

Dave

Our intention was to leverage full posterior distributions to reflect model uncertainty and species covarianceOur goal is to assess whether the GJAM model can reliably detect a known, simulated change in species abundance (e.g., a 20% increase), using full posterior predictive distributions to reflect model uncertainty and species covariance. However, we've encountered persistent challenges when trying to generate these predictions.

1. We initially attempted to use gjamPredict() to generate posterior predictive draws for new covariate conditions (xnew), but repeatedly ran into failures due to internal dependencies and undocumented structure. Even when REDUCT = FALSE, the function appears to rely on latent variable components like fit$inputs$u2s, which are not defined in our models. Patching or bypassing these dependencies has led to additional errors and made the behavior of gjamPredict() difficult to reproduce.
2. To work around this, we began developing a custom manual\_posterior\_predict() function that draws directly from the saved posterior samples (betaBeta, sigmaSave) and simulates predictions for new covariates. This approach is conceptually straightforward — and consistent with standard Bayesian prediction workflows — but in practice, we’ve encountered challenges related to matrix dimensions, reshaping of posterior chains, and uncertainty about how residual covariances are stored when REDUCT = FALSE. As a result, we have not yet been able to generate stable, interpretable posterior predictive distributions for either baseline or simulated-change conditions.
3. **Options Considered**

In light of these issues, we considered a simplified fallback: scaling predicted means by 20% to approximate the effect of increased abundance without re-simulating posterior draws. This would allow us to explore effect sizes more quickly and with less technical overhead.

However, we’re concerned that this shortcut does not propagate posterior uncertainty or capture the joint covariance structure across species. Since our goal is to assess whether a change is statistically **detectable**, not just present in expectation, this approach falls short of supporting robust sensitivity analysis. While it may be useful for preliminary diagnostics, it likely underestimates the role of uncertainty and overstates the model's detection power.

and, suffiI’m writing to ask for your guidance on using the gjam R package for a sensitivity analysis effort focused on NEON plant cover data. We're attempting to assess detectability of year-to-year changes in species percent cover at individual NEON sites using GJAM, but have encountered consistent barriers when working with posterior predictions. I would greatly appreciate your insight into whether the package, especially gjamPredict(), supports our intended workflow.

**Project Overview (Brief)**

We’re modeling percent cover data (type "CA") for vascular plants at individual NEON sites using GJAM. Our goal is to evaluate (1) whether a 20% year-to-year change in species abundance can be detected and (2) how the number of plots sampled affects that detectability.

**Planned Workflow (Logic and Steps)**

1. **Model fitting per site**
   * Fit GJAM with typeNames = "CA" using percent cover of species as responses and two predictors: year (factor) and nlcdClass (land cover, factor).
   * Use REDUCT = FALSE for simplicity and interpretability.
2. **Posterior prediction for new covariates**
   * For a subset of the data or a counterfactual change (e.g., increased cover of a species or shift in year), generate posterior draws of predicted cover across species.
   * Compare predicted draws across years or treatment conditions to quantify detectability of the simulated change.
3. **Sensitivity to sample size**
   * Refit models on reduced plot subsets to evaluate how smaller sample sizes reduce power to detect a known change (simulated via xnew).
4. **Output comparison**
   * Summarize posterior predictions (means, credible intervals) and quantify detection success by comparing distributions across years.

**Key Challenges**

We had hoped to use gjamPredict() to generate posterior predictive samples for new covariate values (xnew), but encountered repeated failures due to undocumented dependencies and internal assumptions. Specifically:

* gjamPredict() fails unless fit$inputs$u2s exists, even when REDUCT = FALSE.
* Even after patching u2s, we encounter errors such as:

typescript

CopyEdit

Error in !attr(u2s, "valid") : invalid argument type

* Attempts to extract posterior draws manually using bgibbs and sgibbs also fail due to shape mismatches (e.g., sigmaSave not reconstructing to S × S covariance matrices).
* The dimensional structure of bgibbs and sgibbs appears flattened, with unclear guidance on reshaping when REDUCT = FALSE.

We attempted to write a custom posterior\_predict\_gjam() function to bypass gjamPredict() entirely and simulate draws directly using the posterior chains. Despite extensive debugging, we could not reliably reconstruct the posterior covariance matrices or obtain stable draws, largely due to dimensional inconsistencies and undocumented output structure.

**Questions**

1. Is gjamPredict() intended to support posterior predictions for xnew when REDUCT = FALSE?
2. Is there a recommended or supported way to generate posterior predictive draws (e.g., ysim) for new data?
3. If the posterior chains (bgibbs, sgibbs) are flattened, is there a reference on how to reshape them appropriately?
4. Are there alternative recommended approaches within gjam for conducting the type of sensitivity simulation we're attempting?

Thank you very much for your time and for developing such a powerful modeling framework. I’d be grateful for any clarification you can provide on these issues.

Slightly different text:  
  
**✅ Current Sensitivity Analysis Workflow: Posterior Prediction Approach**

We fit a single GJAM model per site or plot subset using the observed data.

Then we simulate a 20% increase in species abundance. This simulated change can reflect:

* A shift in predictor variables (e.g., year or land cover class), or
* A direct increase in species values (for exploratory testing).

Using the same model, we apply manual\_posterior\_predict() to generate predictions under both the baseline and modified conditions, drawing across the full posterior distribution of model parameters.

We then assess detectability:

* Are the predicted distributions for the two conditions statistically distinguishable (e.g., non-overlapping credible intervals)?
* How does detection vary with sample size (e.g., number of plots)?

This approach allows us to isolate **prediction uncertainty** from **model estimation uncertainty**, since we do not refit the model for each simulated scenario.

As we continue working through challenges with posterior prediction, we briefly considered an alternative approach: simulating a 20% change by scaling the predicted means from the fitted model. This would allow us to explore effect sizes without additional computation.

However, we’re concerned this shortcut doesn’t propagate posterior uncertainty or reflect the model’s ability to statistically distinguish changes. It also omits species covariance effects, which could influence detectability. While this fallback might be useful for quick diagnostics, we don’t believe it supports robust sensitivity analysis — particularly for our goal of assessing detection thresholds under varying sample sizes. We're continuing to prioritize a full posterior predictive approach but welcome your perspective on trade-offs here.