A high-level question: calculate\_detection\_probability() produces a detection probability for each species and year change combination. please describe what that is

**ChatGPT said:**

Great question — here's a high-level but precise explanation of what calculate\_detection\_probability() is doing in your pipeline:

**🔍 What Is the Detection Probability?**

In this context, **detection probability** is a **posterior probability** that a true change in species percent cover occurred between two years — for a given species at a site — based on predictions from the fitted GJAM model.

**🧠 What It Actually Measures**

For each **species × year pair**:

* You’ve used manual\_posterior\_predict() to generate **two sets of posterior draws**:
  + One set for baseline year (e.g., 2015)
  + One set for changed year (e.g., 2016)
* These represent predicted percent cover across posterior samples under each condition.

Then, for each species:

* You calculate the **difference** in predicted percent cover between years at each posterior draw (i.e., draw\_baseline - draw\_changed).
* You compute the proportion of these differences that are either:
  + Greater than 0 (for increases), or
  + Less than 0 (for decreases),
  + Whichever direction is more dominant.

This proportion is the **detection probability** — i.e.,

**"How likely is it, given the posterior, that this species’ cover changed between these two years — and in a consistent direction?"**

**🧪 Why This Is Meaningful**

It’s not just a test of statistical significance. It's:

* A **Bayesian estimate** of confidence in change.
* Inherently reflects **uncertainty** in both model fit and species covariance.
* More robust than just comparing means — it uses the **full posterior distribution**.

**📈 Example Interpretation**

If for species *STPI2* at site *OSBS* between 2015 and 2016:

* 950 out of 1000 posterior draws show an increase in percent cover,
* Then detect\_prob = 0.95 — you’re 95% confident a real increase occurred.

If:

* Only 550 out of 1000 show a consistent direction,
* Then detect\_prob = 0.55 — weak evidence of change.

**✅ Final Summary**

The detection probability is the **posterior probability that a change in cover occurred in a consistent direction**, for a single species between two years at a site, given the uncertainty estimated by the GJAM model.

Let me know if you'd like a diagram or mathematical notation for this!

**You said:**

but in our case we have simulated or created the 20% change, correct?

**ChatGPT said:**

✅ Yes — **you are absolutely correct**: in *your pipeline*, the **change is not observed**, it is **simulated**.

**🔄 What That Means for calculate\_detection\_probability()**

You’re not estimating the probability that a change occurred in the real world — you're evaluating:

**Given a known (simulated) 20% change in species abundance, how often would the GJAM model detect it?**

So the **detection probability** becomes a measure of **statistical power** — not inference.

**🎯 Restated Definition for *Your Use Case***

In this sensitivity analysis, the detection probability is the **posterior probability that the GJAM model would detect a directionally consistent change in species percent cover**, *assuming a known 20% change was applied*.

It reflects:

* How well the model can "see" a change of that size
* Under a fixed posterior
* Across varying sample sizes (number of plots)
* For each species × year-pair × site

**🧪 What You’re Really Measuring**

This becomes a **simulation-based power analysis**:

* You inject a 20% change into the covariate structure (e.g., shift from year A to year B).
* You check whether the model’s posterior predictions are consistently shifted as a result.
* The detection probability is how often the predicted change is *statistically distinguishable* from no change.

**🔁 Optional Variation**

Later, you could:

* Test 10%, 30%, or variable-sized changes
* Compare species that are more vs less detectable
* Explore how land cover (nlcdClass) affects detectability

Let me know if you’d like a compact explanation suitable for methods text or figures.

**You said:**

summarize the posteriors again for me in this context? what they are and what mean

**ChatGPT said:**

Absolutely — here's a focused summary of what the **posterior predictions** represent in your NEON GJAM sensitivity pipeline:

**🔁 Posterior Predictions in Your Pipeline**

You are working with **posterior predictive distributions** from the GJAM model. For each **species × year condition** (baseline and changed), the model gives you:

A distribution of predicted percent cover values, sampled from the joint posterior of model parameters.

These predictions are generated by your custom function manual\_posterior\_predict() using:

* betaBeta: posterior samples of regression coefficients
* sigmaSave: posterior samples of residual covariance
* xnew: covariate matrix representing either the baseline or the simulated changed condition (e.g., year 2015 vs. year 2016, across a subset of plots)

**📦 The Structure of the Posterior Output**

From manual\_posterior\_predict(), you get:

sql

CopyEdit

Array of shape: [n\_draws, 2, n\_species]

→ Each [draw, condition, species] gives a predicted percent cover

* n\_draws: Number of posterior draws (e.g., 1000)
* 2: Two conditions — baseline vs. simulated change (e.g., 2015 vs 2016)
* n\_species: All species in the site-level model

**🧠 What the Posterior Means**

For any given species:

* You have 1000 draws of predicted percent cover in **baseline** condition (year 1)
* And 1000 draws in **changed** condition (year 2)

Each draw reflects:

* A possible “true” cover value the species might have under that year/condition
* **Given the uncertainty** in both the fitted GJAM model and species covariances

So these posteriors tell you:

"If this 20% change really happened, what cover values would we expect to see for this species — accounting for model uncertainty?"

**🎯 Why They Matter for Detection**

The **overlap** (or separation) between the two posterior distributions tells you:

* Whether the simulated change is *detectable* (statistically distinguishable)
* Detection probability is calculated as the proportion of draws showing a consistent direction of change (positive or negative)

**📈 Example**

For species *RUCU*:

* Baseline year: posterior mean = 4.0%, SD = 1.0%
* Simulated year: posterior mean = 4.8% (i.e., 20% higher), SD = 1.0%
* Detection probability ≈ 0.95 (if most posterior draws show an increase)

If posteriors overlap heavily → detection probability drops

Let me know if you want a visual example or a mathematical representation — these posteriors are the core of your entire sensitivity analysis.

**You said:**

so, to make a determination about reducing sample size we would need to determine a threshold for the posterior predict, right?

**ChatGPT said:**

Yes — you're exactly right again.

To **evaluate the impact of reducing sample size**, you must establish a **threshold** on the detection probability derived from the posterior predictions. This threshold acts as your decision rule for whether a simulated 20% change is **“detectable”**.

**✅ What the Threshold Does**

It tells you:

“Given the posterior predictions for a species × year pair × site × sample size, is the change detectable *enough* to count?”

This turns a continuous detection probability (e.g., 0.91, 0.76, 0.53) into a **binary decision** (yes/no).

**📊 Typical Thresholds**

Here are common thresholds, depending on your desired stringency:

| **Threshold** | **Interpretation** | **Use Case** |
| --- | --- | --- |
| **0.95** | Very confident detection (Bayesian analog to p < 0.05) | High-certainty, publication-grade detection |
| **0.90** | Moderately strong evidence | Policy-relevant signals |
| **0.80** | Reasonable power benchmark | Power analysis / sample size planning |
| **0.70** | Exploratory analysis | Early warning or pilot work |

You could define:

r

CopyEdit

is\_detectable <- detect\_prob >= 0.8

And then ask: *"What proportion of simulations reached detectability at each sample size?"*

**🧠 Why It's Necessary for Sample Size Evaluation**

Without a threshold:

* You can't say whether reducing plots **crosses a boundary** where detection becomes unreliable
* You’d just have a cloud of detection probabilities, but no action-guiding metric

With a threshold:

* You can summarize detection success across replicates (e.g., "85% of simulations with 10 plots reached detectability for RUCU in 2016")
* This allows you to **plot sample size vs detection success** and find optimal tradeoffs

**🔁 Looping Over Thresholds**

Optionally, you could test **multiple thresholds** and report:

* Sample size required to exceed 80% detection at 90% certainty
* Sensitivity of results to choice of threshold

This is common in **simulation-based power analysis** frameworks.

Let me know if you’d like:

* A function to classify detectability given a threshold
* Summary tables or figures showing sample size vs detection success across species or sites

You're doing everything right — this is exactly the next step in applying your posteriors.

**You said:**

and summarize again how we imposed the 20% increase without refitting the model

**ChatGPT said:**

Absolutely — here’s a clear summary of **how your pipeline imposes a simulated 20% increase in species abundance without refitting the GJAM model**, which is critical to preserving the posterior uncertainty structure.

**🔁 The Key Principle**

You **do not** change the observed data or refit the model.  
Instead, you simulate how the model would respond to a **change in the predictors**, using the fixed posterior from the full model.

**🧠 How the 20% Change Is Imposed**

1. **You start with a fully fitted GJAM model**  
   This model includes:
   * Posterior draws for regression coefficients (betaBeta)
   * Posterior residual covariance (sigmaSave)
   * Fixed species covariance structure based on *all plots*
2. **You construct xnew — a new design matrix**  
   You build two versions of xnew:
   * One for the **baseline year** (e.g., 2015)
   * One for the **changed year** (e.g., 2016)

These matrices come from:

* + A **subset of plots** (to simulate reduced sampling)
  + With only the **year predictor changed**, keeping everything else constant

Example:

r

CopyEdit

xnew\_baseline <- xdata\_subset %>% mutate(year = "2015")

xnew\_changed <- xdata\_subset %>% mutate(year = "2016")

xnew <- bind\_rows(xnew\_baseline, xnew\_changed)

⚠️ You do **not** manually add a +20% to the response variable. Instead:

1. **The GJAM model interprets the change in year as a covariate effect**  
   When you run:

r

CopyEdit

manual\_posterior\_predict(fit, xnew)

The model uses betaBeta to generate predictions for each condition:

* + For year = 2015
  + For year = 2016

If the model’s year effect implies a **20% increase** in predicted cover for a species, that comes from the structure of the model and the way the xnew matrix triggers changes in predicted values.

1. **This setup reflects a counterfactual simulation**
   * You’re not asking “what happened?”
   * You’re asking: *“If there were a 20% increase in abundance due to year effect, would the model detect it?”*

**🧮 Why No Manual “+20%” Scaling?**

Because:

* You are relying on the model’s internal mappings (via betaBeta) between year and abundance.
* You **impose** the 20% change **by simulating a covariate shift**, not manipulating outcomes.
* This retains the correct posterior variance and covariance across species and across conditions.

**✅ Summary Statement**

The 20% change is imposed by modifying the **year** covariate in the new predictor matrix (xnew) for a fixed subset of plots.  
Predictions from the full GJAM posterior (betaBeta, sigmaSave) simulate how species abundances would respond to that change — *without ever refitting the model*.

Let me know if you'd like a diagram or a code snippet wrapper for this approach — it's the cornerstone of your simulation-based power analysis.

Top of Form

Tools

Bottom of Form

ChatGPT can make mistakes. Chec