McElreath Chapter 13: Models with Memory

Mike Wolfe

08/05/2021

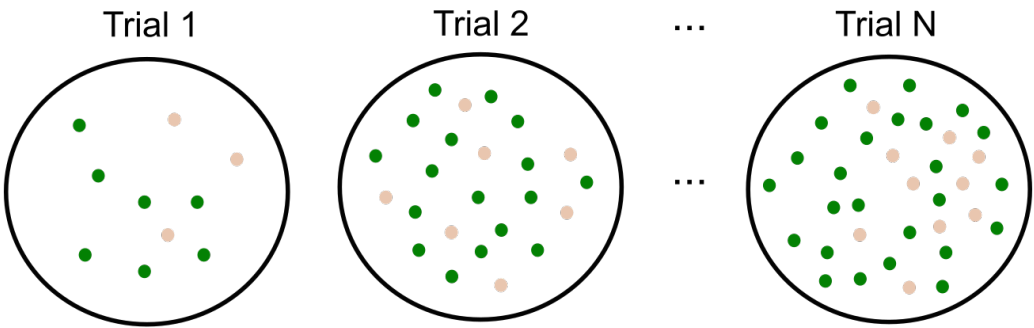
A motivating example - Cafe waiting times

- You want to estimate how much time you have to wait for your coffee at any given cafe in the city
- Let's say you start with a vague Gaussian prior of $\mu = 5$ min and $\sigma = 1$ min
- You visit a cafe and get your coffee in 4 minutes, which you use to update your expectations
- You then visit a second cafe. What to use as your prior?
  - Could use the posterior after the second cafe, but assumes that this cafe has the same average waiting time
- We don't want to assume that all cafes are exactly the same
- We also don't want to completely ignore the information about waiting times from the first cafe
- How can we do better?

Multilevel models

- Instead let's try to estimate the overall average of waiting times as well as the variation between cafes
- We will use this distributional information to inform our individual cafe estimates
- This allows us to **pool** information between cafes without assuming they are all the same
- Benefits
  - Improved estimates for repeat sampling
  - Improved estimates for imbalanced sampling
  - **Explicit Estimates of variation** between groups
  - Avoidance of averaging to preserve variation

A colony counting example

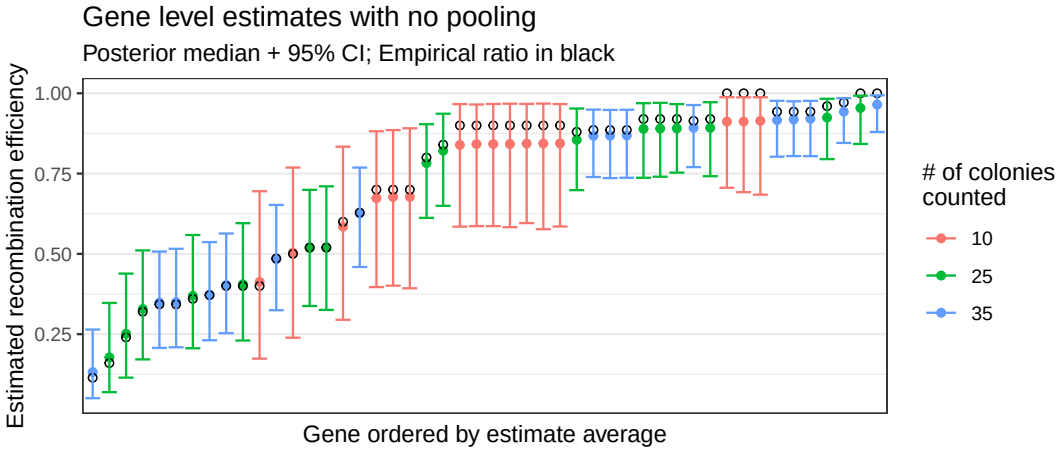


Estimating recombination efficiency - No pooling

- First we are going to treat every targeted gene as it's own unique case
- This is a straight forward application of the models we learned for count data
- No pooling of information between genes will be used here
- We will fit a simple binomial model with an intercept for each gene:

$$
\begin{aligned}
\text{gfp}_i &\sim \text{Binomial}(n_i, p_i) \\
\text{logit}(p_i) &= \alpha_{gene[i]} \\
\alpha_j &\sim \text{Normal}(0, 1.5)
\end{aligned}
$$

Estimating recombination efficiency - No pooling



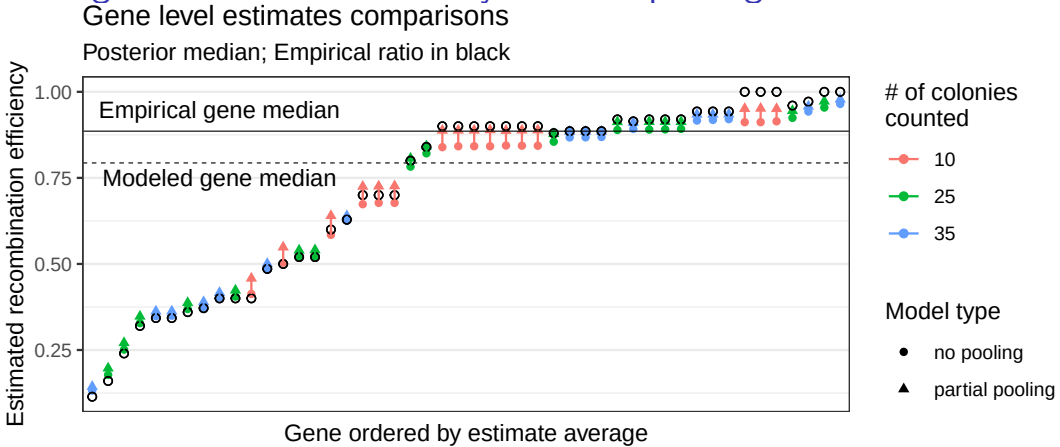
Estimating recombination efficiency - Partial pooling

- Now we want to use a multilevel model to allow for **partial pooling**
- In other words, we want the information about one gene help us inform our estimate about another gene
- The easiest way to do this is to treat the $\alpha_{\text{gene}[i]}$ estimates as normally distributed with some overall average $\bar{\alpha}$ and standard deviation as σ.
- We will then simultaneous learn each estimate $\alpha_{\text{gene}[i]}$ and the overall population-level estimates $\bar{\alpha}$ and σ from the data.
- The model looks like this

$$
\begin{aligned}
\text{gfp}_i &\sim \text{Binomial}(N_i, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{gene}[i]} \\
\alpha_j &\sim \text{Normal}(\bar{\alpha}, \sigma) \\
\bar{\alpha} &\sim \text{Normal}(0, 1.5) \\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

- Where $\bar{\alpha}$ and σ are **hyperparameters** and their priors are **hyperpriors**

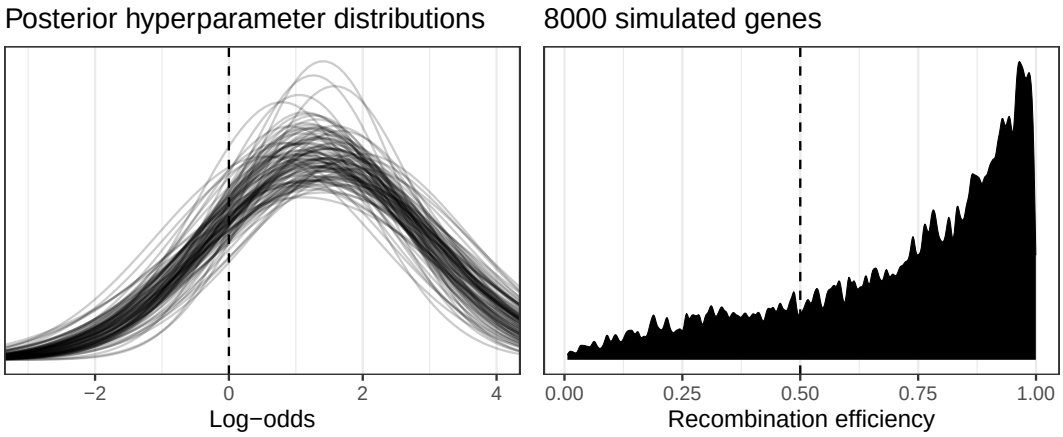
Estimating recombination efficiency - Partial pooling



- Here partial pooling moves the estimates towards the learned average estimate relative to the empirical ratio
- However, compared to the simpler model with no pooling, some uncertain high estimates are enabled to move closer to their empirical estimates

Thinking about and interpreting the hyperparameters

- We can use the posterior distribution of the hyperparameters to reason about what to expect when targeting a future gene
- The advantage of the multilevel model is that we aren't assuming every gene is the same, rather, we are **explicity** modeling the variation between genes

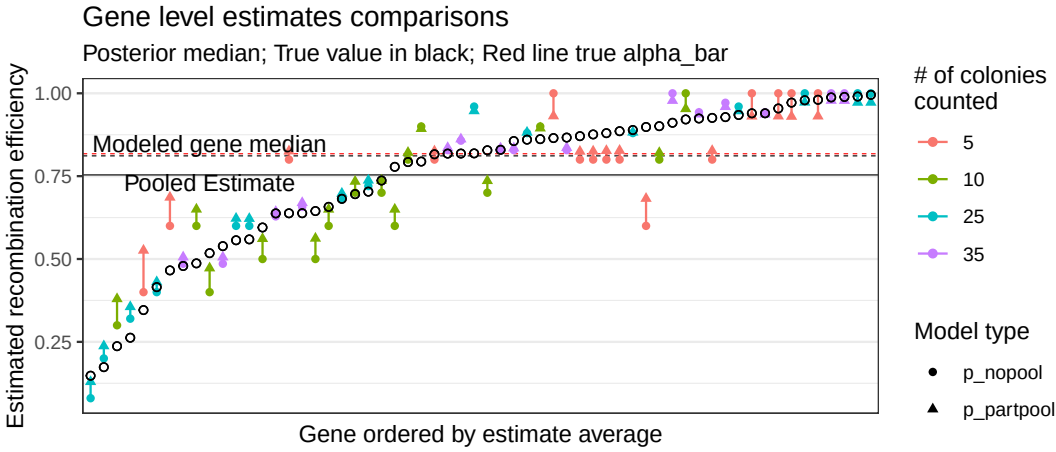


Do multilevel models really help that much?

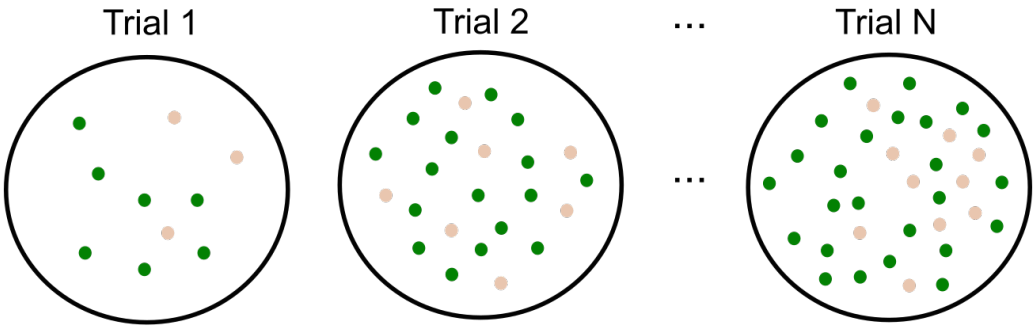
- Simulate data where we know what the ground truth is
- Compare three different approaches to estimate the true parameters
  - **Complete pooling** - every gene is identical. Estimate one intercept
  - **No pooling** - every gene is different. Estimate each intercept seperately
  - **Partial pooling** - genes are different but knowing something about one tells you something about the other. Use the adaptive regularization learned from the data
- We are going to simulate data from a model identical to the one we used before

$$
\begin{aligned}
\text{gfp}_i &\sim \text{Binomial}(N_i, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{gene}[i]} \\
\alpha_j &\sim \text{Normal}(\bar{\alpha}, \sigma) \\
\bar{\alpha} &\sim \text{Normal}(0, 1.5) \\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

Multilevel models improve estimates on average



Can we deal with more than one grouping?

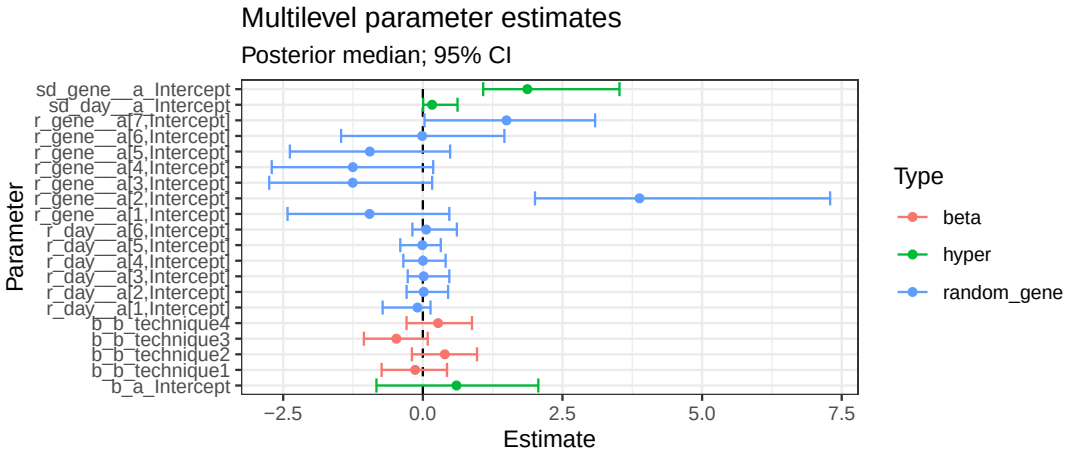


Modeling variation in different groupings using multiple varying intercepts

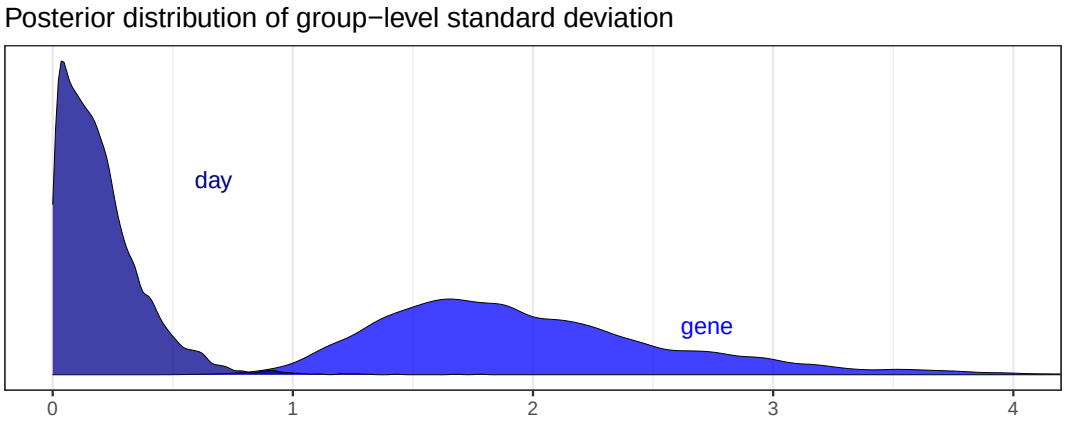
- This time we gathered replicates on different days
- We also tried to obtain the desired change using different techniques

$$
\begin{aligned}
\text{gfp}_i &\sim \text{Binomial}(n_i = 1, p_i) \\
\text{logit}(p_i) &= \alpha_{\text{gene}[i]} + \gamma_{\text{day}[i]} + \beta_{\text{technique}[i]} \\
\beta_j &\sim \text{Normal}(0, 0.5), \text{ for } j = 1 \ldots 4 \\
\alpha_j &\sim \text{Normal}(\bar{\alpha}, \sigma_\alpha), \text{ for } j = 1 \ldots 7 \\
\gamma_j &\sim \text{Normal}(0, \sigma_\gamma), \text{ for } j = 1 \ldots 6 \\
\bar{\alpha} &\sim \text{Normal}(0, 1.5) \\
\sigma\alpha &\sim \text{Exponential}(1) \\
\sigma\gamma &\sim \text{Exponential}(1)
\end{aligned}
$$

Multiple varying intercepts



Estimating variance from different sources



How do I make predictions with all of this?

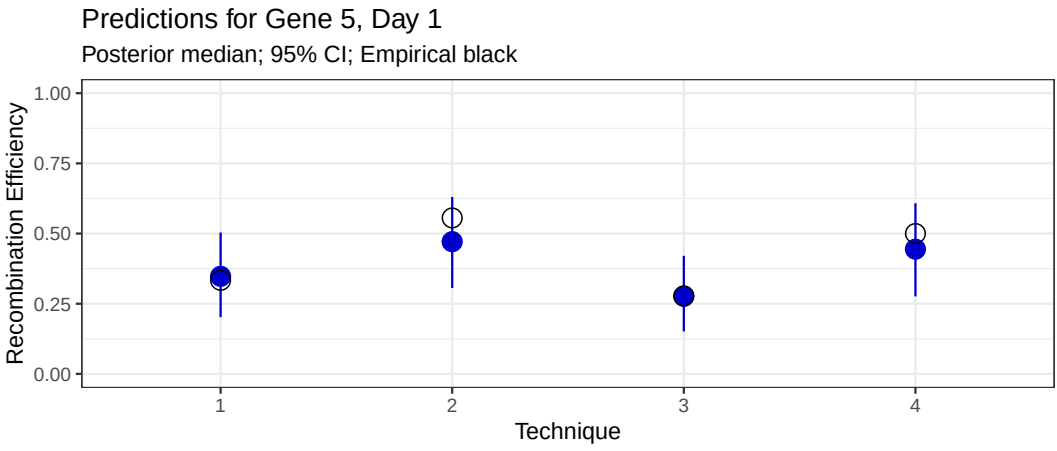
- There are a couple of different scenarios you could consider:
  - Making predictions for the same clusters you have already seen
  - Making predictions for new clusters when thinking only about the average effects
  - Making predictions for new clusters when thinking about variation from clusters

Making predictions for a set of cluster values

- Predicting for a known set of clusters is as simple as doing the math
- For example for Gene 5, day 1:

$$\text{inv logit}(\alpha_5 + \gamma_1 + \beta_i) \text{ for } i = 1 \ldots 4$$

Making predictions for a set of cluster values

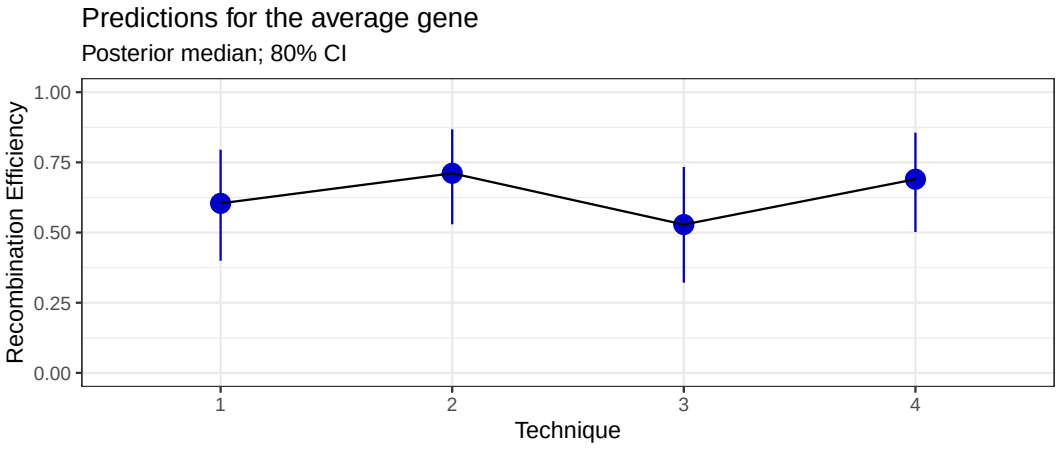


Making predictions for new clusters - the average gene

- First let's think about making a prediction for the average gene
- Here we will make good use of the $\bar{\alpha}$ parameter
- Since the average effect of the day was zero we will ignore it
- Calculating the effects for the average gene is then

$$\text{inv logit}(\bar{\alpha} + \beta_i) \text{ for } i = 1 \ldots 4$$

Making predictions for new clusters - the average gene

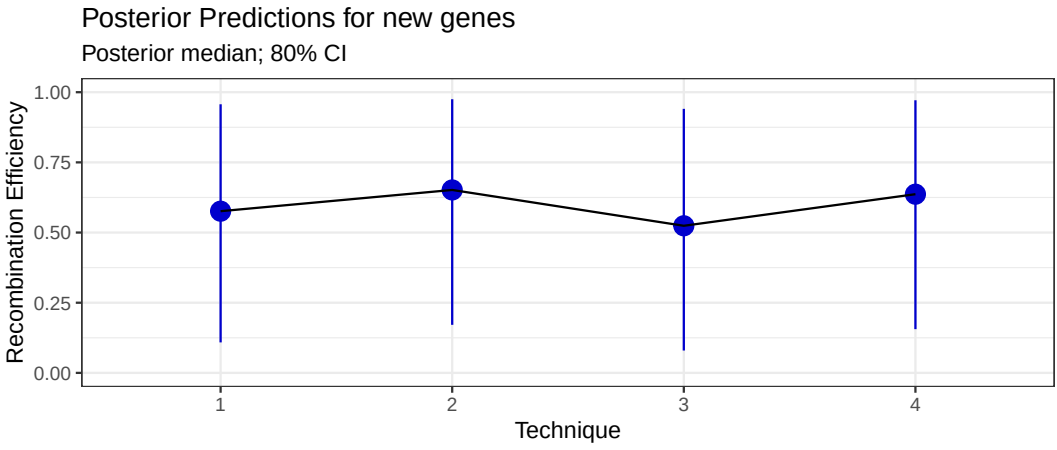


Making predictions for new clusters - including gene-to-gene variation

- Next let's also consider the variation between different genes in our estimate
- This time we will need to simulate from two levels in the model

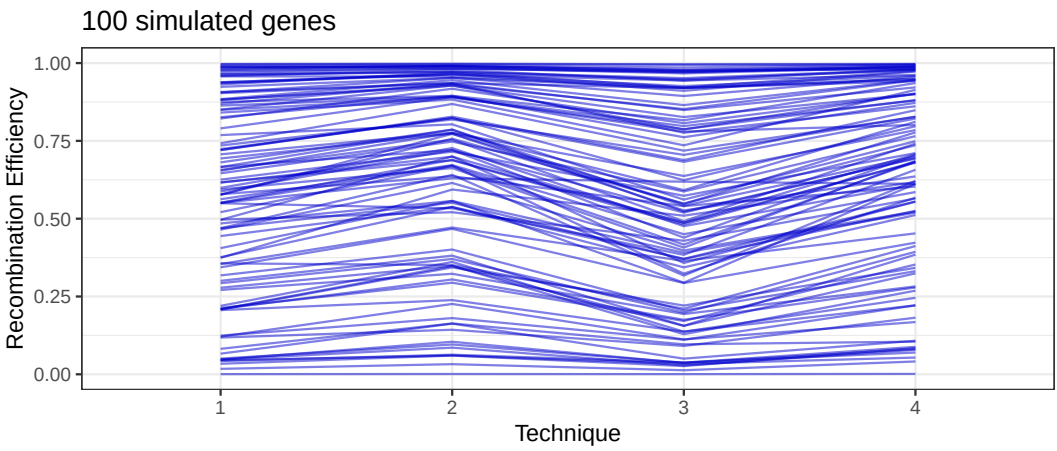
$$\alpha_{\text{sim}} \sim \text{Normal}(\bar{\alpha}, \sigma_{\text{gene}})$$
$$\text{inv logit}(\alpha_{\text{sim}} + \beta_i) \text{ for } i = 1 \ldots 4$$

Making predictions for new clusters - including gene-to-gene variation

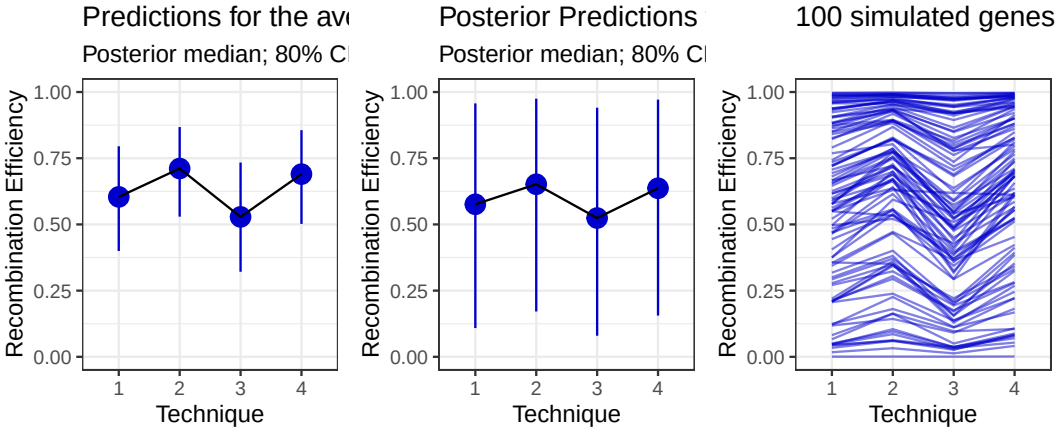


Making predictions for new clusters - Simulation

- Instead of looking at summaries let's actually look at a bunch of simulated genes



Making predictions for new clusters - Summary



Multilevel models - Major take aways and a look ahead

- Multilevel models help you explicitly model variation based on groupings in your data
- The partial pooling aspect of these models allows you to share information between related groups. This can improve estimates where you have sparse sampling
- So far we have focused on **varying intercepts** models where we only use the multilevel approach for intercepts in our models
- We saw that we could model variation for more than one type of group in a dataset
- In the next chapters we will see how we can extend this approach to allow for
  - Varying slopes
  - Dealing with continuous "groups" with a Gaussian process
  - Dealing with missing data
  - Using these techniques with more domain specific models such as DiffEQs