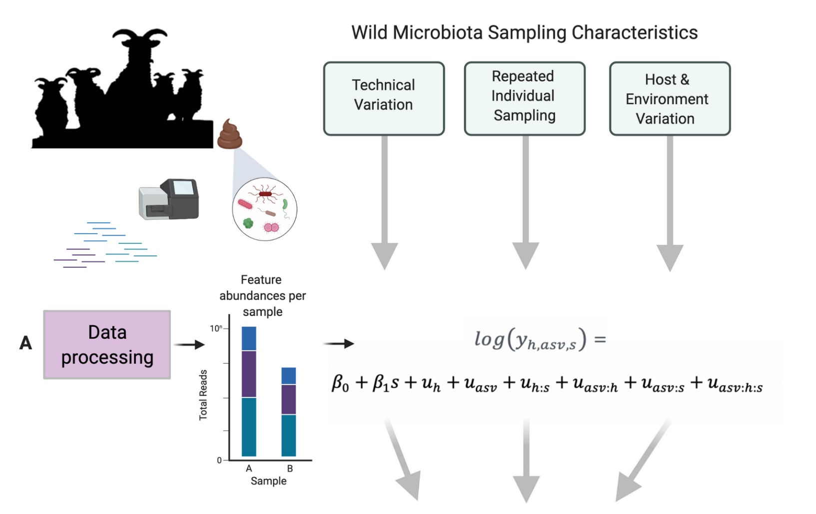
Bayesian multilevel modeling of phenotypic plasticity

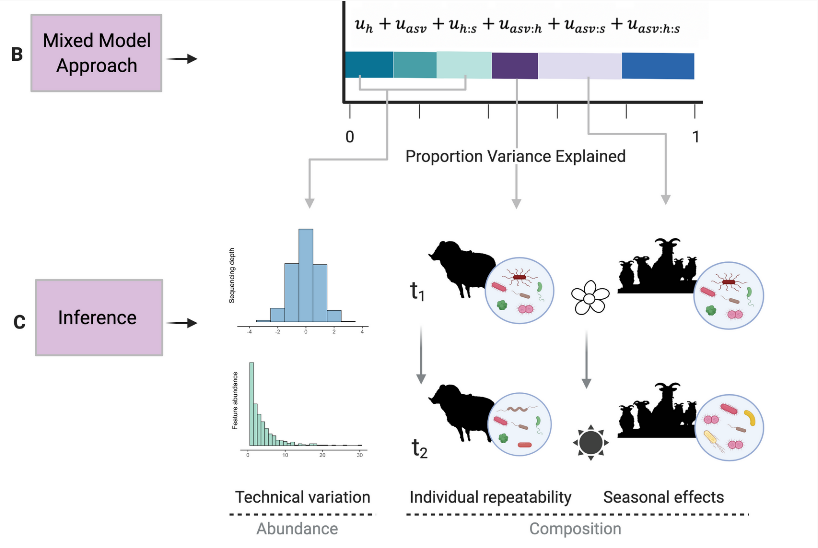
Paul Bürkner, TU Dortmund University

Example: Wild metabarcoding data (1)

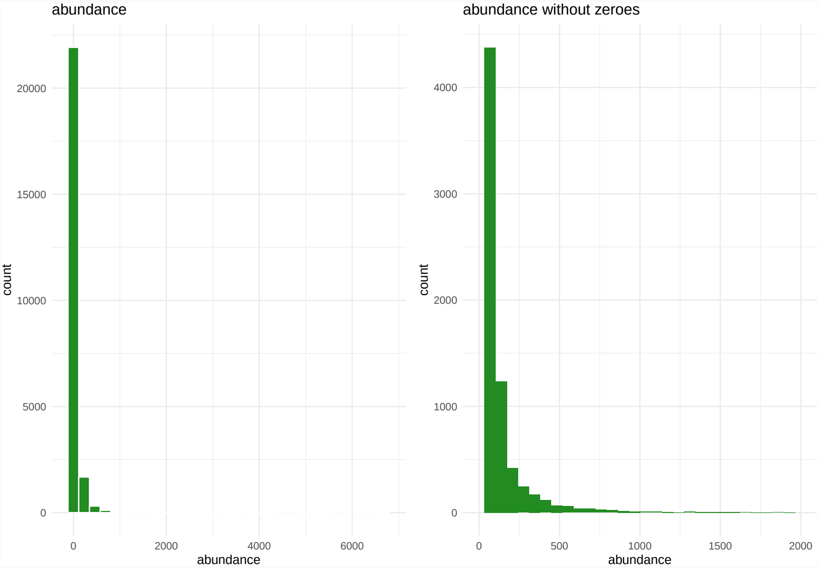
Reference: Sweeny et al. (2023). doi:10.1128/msystems.00040-23



Example: Wild metabarcoding data (2)



Response data overview



Example: Bayesian multilevel model for many groups

```r
library(brms)

bform_ab_zinb <- bf(
  abundance ~ 1 + (1|sample) + (1|asv) +
    (1|sex) + (1|age_class) + (1|age_class:asv),
  family = "zero_inflated_negbinomial"
)

fit_ab_zinb <- brm(
  bform_ab_zinb,
  data = data_metabar,
  ...
)
```

How to specify priors in multilevel models?

Explained variance in linear models

$$y_n \sim \text{normal}(\mu_n, \sigma)$$
$$\mu_n = b_0 + \sum_{k=1}^{K} b_k x_{kn}$$
$$b_k \sim \text{normal}(0, \lambda_k)$$

Implied percentage of explained variance:

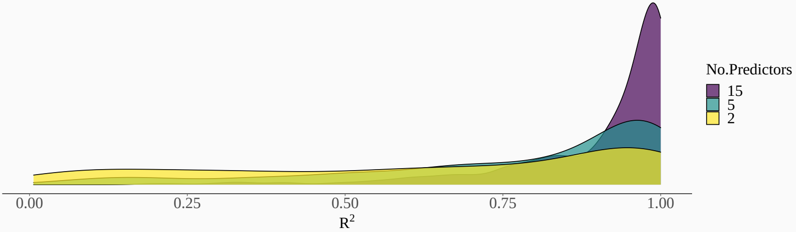
$$R^2 = \frac{\text{var}(\mu)}{\text{var}(\mu) + \sigma^2},$$

Assuming standardized predictors:

$$\text{var}(\mu) = \sum_{k=1}^{K} \lambda_k^2.$$

Priors implied on R^2

For constant-variance priors on b, the implied R^2 increases as the number of predictors increases.



R2D2 priors for simple linear models

Turn around the process and define a prior on R^2, which then implies a prior on the coefficients b.

$$b_k \sim \text{Normal}(0, \frac{\sigma^2}{\sigma_k^2}\tau^2\phi_k)$$
$$\sigma_k^2 = \text{Var}(x_k)$$

Priors on the hyperparameters:

$$R^2 \sim \text{Beta}(\mu_{R^2}, \varphi_{R^2})$$
$$\tau^2 = \frac{R^2}{1 - R^2} \sim \text{BetaPrime}(\mu_{R^2}, \varphi_{R^2})$$
$$\phi \sim \text{Dirichlet}(\alpha)$$

Reference: Zhang et al. (2022). doi:10.1080/01621459.2020.1825449

R2D2 priors for linear multilevel models

Linear multilevel model with varying intercepts and slopes:

$$
\begin{aligned}
y_n &\sim \text{Normal}(\mu_n, \sigma^2) \\
\mu_n &= b_0 + \sum_{k=1}^{K} x_{kn} b_k + \sum_{g \in G_0} u_{0 g_{j[n]}} + \sum_{k=1}^{K} x_{kn} \left(\sum_{g \in G_k} u_{k g_{j[n]}} \right) \\
b_k &\sim \text{Normal}\left(0, \frac{\sigma^2}{\sigma_k^2} \tau^2 \phi_k \right) \quad k = 1, \dots, K \\
u_{k g_j} &\sim \text{Normal}\left(0, \frac{\sigma^2}{\sigma_k^2} \tau^2 \phi_{kg} \right) \quad k = 0, \dots, K \\
\tau^2 &= \frac{R^2}{1 - R^2} \\
R^2 &\sim \text{Beta}(\mu_{R^2}, \varphi_{R^2}), \;\; \phi \sim \text{Dirichlet}(\alpha)
\end{aligned}
$$

Reference: Aguilar & Bürkner (2023). doi:10.1214/23-EJS2136

R2 priors of generalized linear models

$$y_n \sim \text{Likelihood}(\mu_n, ...)$$

$$\mu_n = \text{inverseLink}(b_0 + \sum_{k=1}^{K} b_k x_{kn})$$

$$b_k \sim \text{Normal}\left(0, \frac{\hat{\sigma}^2}{\sigma_k^2}\tau^2\phi_k\right)$$

Challenges:

- How to define/compute $\hat{\sigma}^2$
- How to interpret R^2 on the link scale

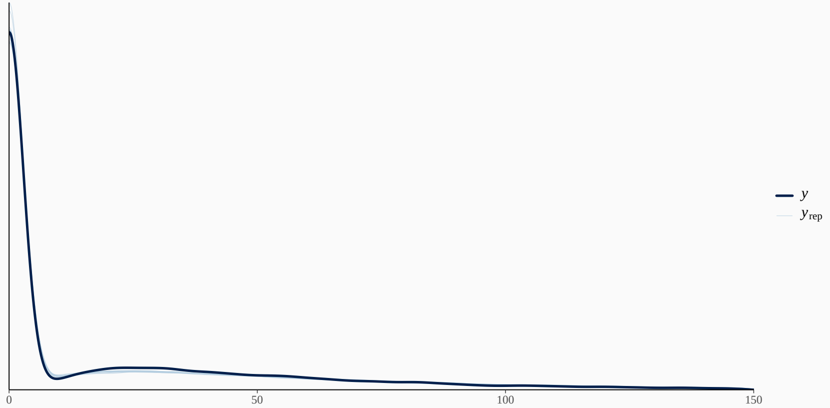
Zero-inflated negative binomial model

```r
bprior_ab_R2 <- prior(
  R2D2(mean_R2 = 0.5, prec_R2 = 2,
       cons_D2 = 10, main = TRUE),
  class = "sd"
)
```

```r
fit_ab_zinb_R2 <- brm(
  bform_ab_zinb,
  data = data_metabar,
  prior = bprior_ab_R2,
  ...
)
```

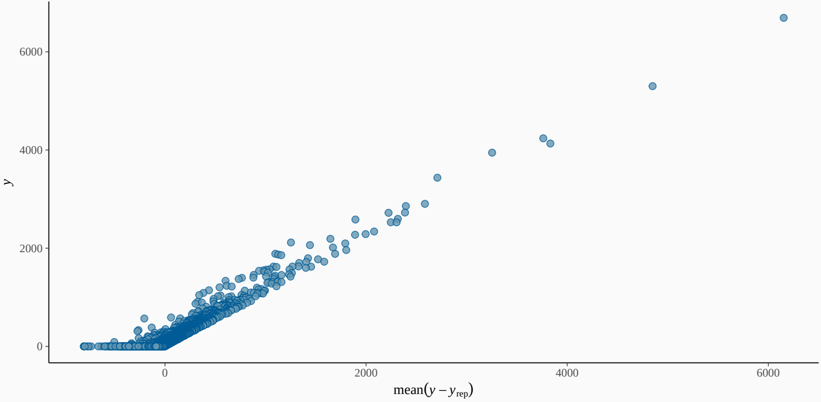
Posterior predictive checks: Marginal

```
pp_check(fit_ab_zinb_R2) + xlim(c(0, 150))
```



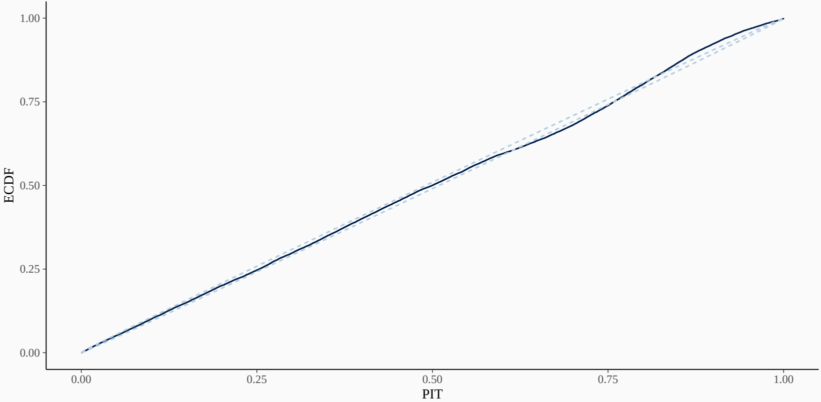
Posterior predictive checks: Residual

```
pp_check(fit_ab_zinb_R2, type = "error_scatter_avg")
```

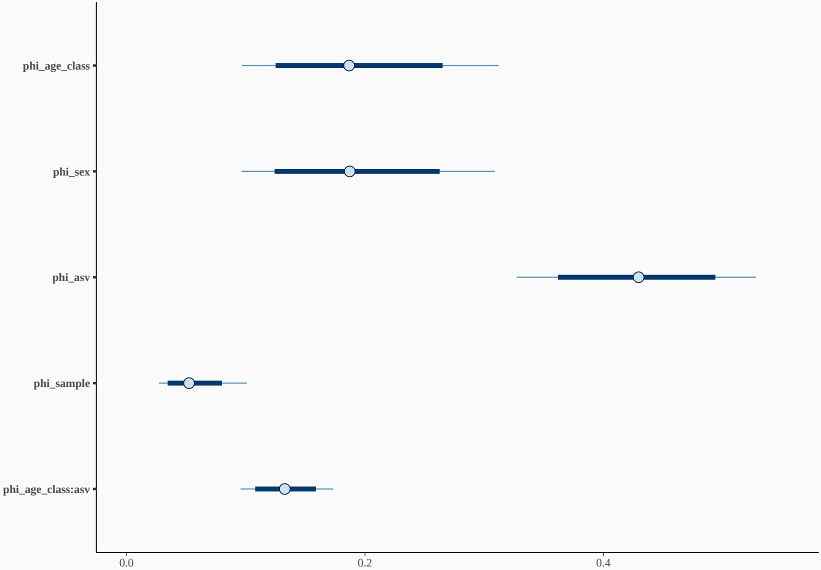


Posterior predictive checks: ECDF rank scores

```
pp_check(fit_ab_zinb_R2, type = "pit_ecdf")
```



Percentages of explained variance



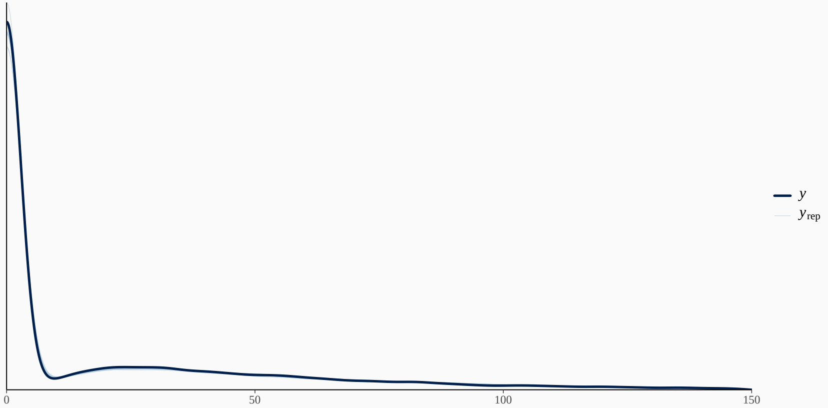
Zero-inflated Poisson model

```
bform_ab_zip <- bf(
  abundance ~ 1 + (1|sex) + (1|age_class) +
    (1|sample) + (1|asv) + (1|age_class:asv) + (1|obs),
  family = "zero_inflated_poisson"
)
```

```
fit_ab_zip_R2 <- brm(
  bform_ab_zip,
  data = data_metabar,
  prior = bprior_ab_R2,
  ...
)
```

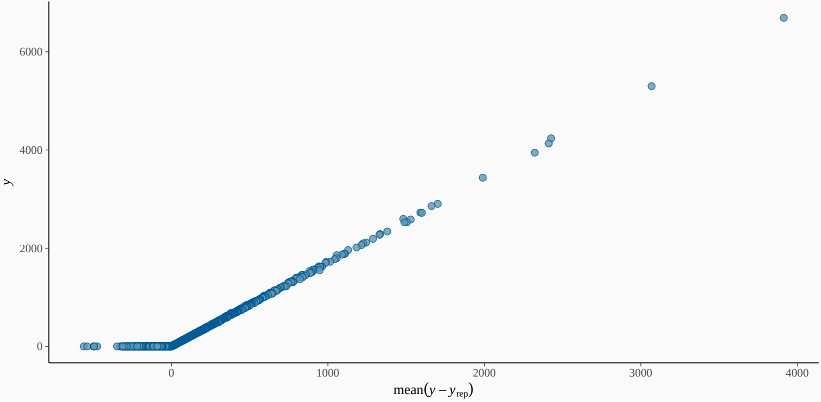
Posterior predictive checks: Marginal

```
pp_check(fit_ab_zip_R2) + xlim(c(0, 150))
```



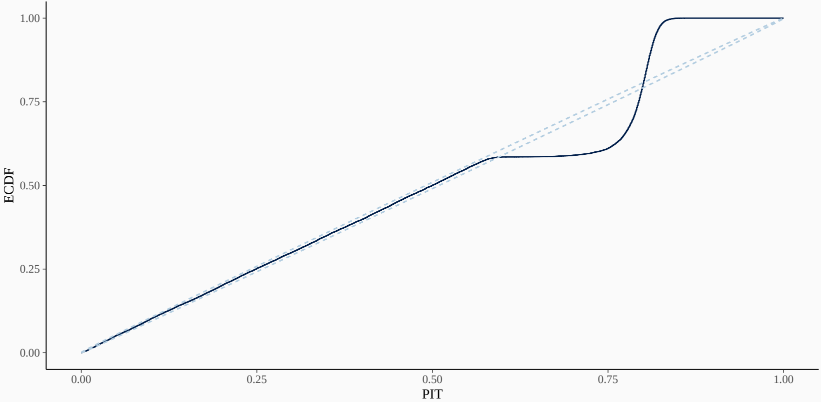
Posterior predictive checks: Residual

```
pp_check(fit_ab_zip_R2, type = "error_scatter_avg")
```

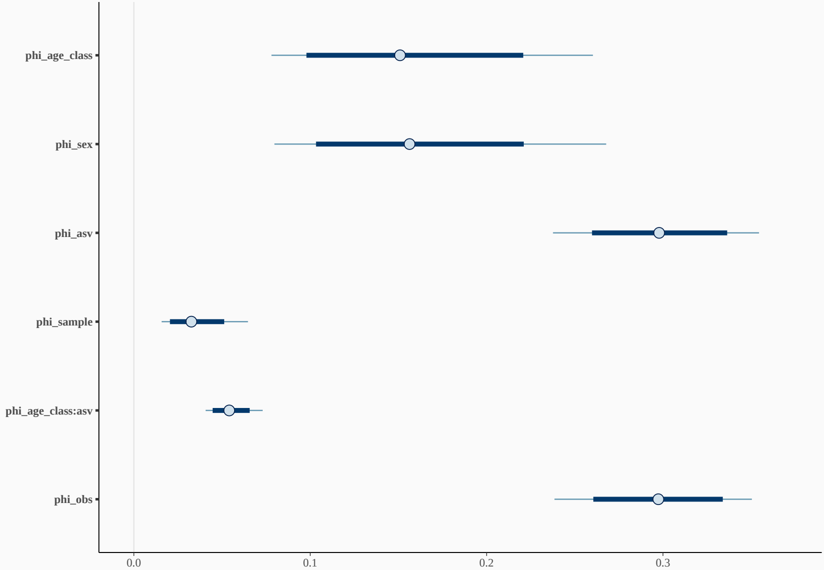


Posterior predictive checks: ECDF rank scores

```
pp_check(fit_ab_zip_R2, type = "pit_ecdf")
```



Percentages of explained variance



LOO-CV for a zero-inflated negative binomial model

```r
loo(fit_ab_zinb_R2)
```

```
Computed from 8000 by 24150 log-likelihood matrix.

         Estimate    SE
elpd_loo -68196.1 446.2
p_loo      1369.8  43.4
looic    136392.3 892.3
------
MCSE of elpd_loo is NA.
MCSE and ESS estimates assume MCMC draws (r_eff in [0.3, 2.6]).

Pareto k diagnostic values:
                         Count Pct.    Min. ESS
(-Inf, 0.7]   (good)     23988 99.3%   66
   (0.7, 1]   (bad)        120  0.5%   <NA>
   (1, Inf)   (very bad)    42  0.2%   <NA>
See help('pareto-k-diagnostic') for details.
```

LOO-CV for our zero-inflated Poisson model

```r
loo(fit_ab_zip_R2)
```

```
Computed from 4000 by 24150 log-likelihood matrix.

         Estimate    SE
elpd_loo -57331.8 343.6
p_loo      7795.1  63.7
looic    114663.6 687.3
------
MCSE of elpd_loo is NA.
MCSE and ESS estimates assume MCMC draws (r_eff in [0.1, 1.3]).

Pareto k diagnostic values:
                         Count Pct.    Min. ESS
(-Inf, 0.7]   (good)     15075 62.4%   34
   (0.7, 1]   (bad)       7562 31.3%   <NA>
   (1, Inf)   (very bad)  1513  6.3%   <NA>
See help('pareto-k-diagnostic') for details.
```

Comparison of the two models

```
loo_compare(fit_ab_zip_R2, fit_ab_zinb_R2)
```

```
               elpd_diff se_diff
fit_ab_zip_R2        0.0       0.0
fit_ab_zinb_R2 -10864.4     134.2
```

Issues with zero-inflated models

Zeros that are modeled by the zero-inflated arm via a constant zero-inflation parameter are “lost” for the predictive model part.

- Is there a separate mechanism that implies the extra zeros?
- Do we want to model this mechanism?

Potential solutions:

- Remove zeros if you have good reasons for doing so
- Predict the zero-inflation parameter
- Treat the zeros as part of the regular model even if that leads to some misspecification

Why are my chains are not mixing?

Model structure not identified by the data:

- Too few groups: Model as fixed effects (or much stronger priors)
- Too few observations per group: Reduce multilevel complexity (or much stronger priors)
- Nearly colinear model terms: Remove redundant terms (or much stronger priors)
- Leapfrog step size too large: Increase `adapt_delta`

Too little warmup or post-warmup draws:

- Warmup of 1000 should suffice
- Post-warmup iterations of a few 1000s should suffice

Learn More

Contact details:

- Website: https://paulbuerkner.com/
- Email: paul.buerkner@gmail.com
- Bluesky: @paulbuerkner.com

brms:

- Website: https://paulbuerkner.com/brms
- Github: https://github.com/paul-buerkner/brms
- Book (draft): https://paulbuerkner.com/software/brms-book