glmmTMB balances speed and flexibility among packages for Zero-inflated Generalized Linear Mixed Modeling

Brooks, Mollie Elizabeth; Kristensen, Kasper; van Benthem, Koen J.; Magnusson, Arni; Berg, Casper Willestofte; Nielsen, Anders; Skaug, Hans J.; Machler, Martin; Bolker, Benjamin M.

Published in:
The R Journal

Link to article, DOI:
10.32614/RJ-2017-066

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

glmmTMB Balances Speed and Flexibility Among Packages for Zero-inflated Generalized Linear Mixed Modeling

by Mollie E. Brooks, Kasper Kristensen, Koen J. van Benthem, Arni Magnusson, Casper W. Berg, Anders Nielsen, Hans J. Skaug, Martin Mächler, Benjamin M. Bolker

Abstract Count data can be analyzed using generalized linear mixed models when observations are correlated in ways that require random effects. However, count data are often *zero-inflated*, containing more zeros than would be expected from the typical error distributions. We present a new package, **glmmTMB**, and compare it to other R packages that fit zero-inflated mixed models. The **glmmTMB** package fits many types of GLMMs and extensions, including models with continuously distributed responses, but here we focus on count responses. **glmmTMB** is faster than **glmmADMB**, **MCMCglmm**, and **brms**, and more flexible than **INLA** and **mgcv** for zero-inflated modeling. One unique feature of **glmmTMB** (among packages that fit zero-inflated mixed models) is its ability to estimate the Conway-Maxwell-Poisson distribution parameterized by the mean. Overall, its most appealing features for new users may be the combination of speed, flexibility, and its interface's similarity to **lme4**.

Introduction

Observed response variables are often in the form of discrete count data, e.g., the number of times that owl nestlings beg for food (Roulin and Bersier, 2007), counts of salamanders in streams (Price et al., 2016), or counts of parasite eggs in fecal samples of sheep (Denwood et al., 2008). These counts are often analyzed using generalized linear models (GLMs) and their extensions (O'Hara and Kotze, 2010; Wilson and Grenfell, 1997). GLMs quantify how expected counts change as a function of predictor variables, e.g., nestlings change their behavior depending on which parent they interact with (Roulin and Bersier, 2007), salamander abundance decreases in streams affected by coal mining (Price et al., 2016), and helminth infection intensity in sheep decreases in response to treatment with anthelmintic drugs (Wang et al., 2017). Repeated measurements on the same individual, at the same location, or observations during the same time period are often correlated; this correlation can be accounted for using random effects in generalized linear mixed models (GLMMs; Bolker et al., 2009; Bolker, 2015).

These types of count data are commonly modeled with GLMs and GLMMs using either Poisson or negative binomial distributions. For the Poisson distribution, the variance is equal to the mean. When data are overdispersed — meaning the variance is larger than the mean — they are often instead modeled using the negative binomial distribution, which can be defined as a mixture of Poisson distributions with Gamma-distributed rates. Overdispersion can also be addressed by adding a random effect with one level for each observation, i.e., a log-normal Poisson distribution (Elston et al., 2001; Hadfield, 2010; Harrison, 2014, 2015). Ignoring overdispersion causes confidence intervals to be too narrow and inflates the rate of false positives in statistical tests (Rhodes, 2015). When data are either over- or underdispersed, they can be modeled with the lesser-known, Conway-Maxwell-Poisson distribution (Shmueli et al., 2005; Lynch et al., 2014; Barriga and Louzada, 2014). Depending on the dispersion, the upper tail of the Conway-Maxwell-Poisson distribution can be either longer or shorter than that of the Poisson (Sellers and Shmueli, 2010). With two parameters, the Conway-Maxwell-Poisson is a generalization of the Poisson distribution and, depending on the dispersion parameter, it also includes the Bernoulli and geometric distributions as special cases (Sellers and Shmueli, 2010). Among other generalizations, the Sichel and Delaporte distributions (Stasinopoulos et al., 2017) provide flexibility in skewness in addition to dispersion.

For these distributions, the expected number of zeros decreases as the mean increases. However, when multiple processes underlie the observed counts, the counts can contain many zeros even if the mean is much greater than zero. For example, an observation of a stream with zero salamanders could be a "structural" zero due to the stream being uninhabitable due to mining waste, or a "sampling" zero due to the combination of a low mean (due to poor ecological suitability and/or low detectability) and sampling variation (Price et al., 2016). Zero-inflated (more broadly zero-altered) GLMs allow us to model count data using a mixture of a Poisson, negative binomial, or Conway-Maxwell-Poisson distribution and a structural zero component. Models that ignore zero-inflation, or attempt to handle it in the same way as simple overdispersion, can yield biased parameter estimates (Harrison, 2014).

In this article, we outline the R packages available for fitting models to count data while introducing **glmmTMB**. We assume that the reader already has a basic understanding of GLMs (Buckley, 2015),

GLMMs (Bolker, 2015), and zero-altered models (Zeileis et al., 2008; Harrison, 2014)).

Several R packages are available for fitting zero-inflated models: **pscl**, **INLA**, **MCMCglmm**, **glmmADMB**, **mgcv**, **brms**, and **gamlss** (Table 1; Zeileis et al., 2008; Rue et al., 2009; Hadfield, 2010; Skaug et al., 2012; Wood et al., 2016; Bürkner, 2017; Stasinopoulos et al., 2017). The widely-used **pscl** package can fit zero-inflated and hurdle GLMs with predictor variables on the zero-inflation using maximum likelihood estimation (MLE: Zeileis et al., 2008). For example, **pscl** can be used to test the hypothesis that sheep fecal egg counts depend on age and structural zeros depend on genotype. However, **pscl** cannot model the correlation within sampling units caused by repeated samples; this requires random effects. Omitting random effects and thereby ignoring correlation makes statistical inference anti-conservative (Bolker et al., 2009; Bolker, 2015). Several other packages have similar capabilities for fitting zero-inflated GLMs (**flexmix**, **MXM**, **VGAM**: Grün and Leisch, 2008; Lagani et al., 2017; Yee, 2017), but in this paper we focus on packages that can also estimate random effects. One such package is **glmmADMB** which can fit zero-inflated GLMMs (Skaug et al., 2012). However, **glmmADMB** cannot fit models where the degree of zero-inflation varies across observational units; thus, it is only appropriate for models where all observational units have an equal probability of producing a structural zero. **INLA** has the same limitation as **glmmADMB** (Rue et al., 2009). The **mgcv** package can only fit zero-inflated GLMMs with predictors of zero-inflation when using a Poisson distribution (Wood et al., 2016). The **MCMCglmm** and **brms** packages can fit zero-inflated GLMMs with predictors of zero-inflation, but they are relatively slow (as we will show) because they rely on Markov chain Monte Carlo (MCMC) sampling (Bürkner, 2017; Hadfield, 2010). **gamlss** is a flexible package that fits generalized additive models with predictors on all parameters of a distribution; its scope includes several zero-inflated and zero-altered distributions (Stasinopoulos et al., 2017).

The list of features documented here is not exhaustive. It should be appreciated that **brms**, **gamlss** and **MCMCglmm** have additional features that go beyond the scope of zero-inflated GLMMs (Bürkner, 2017; Stasinopoulos et al., 2017; Hadfield, 2010). We focus on the process of fitting models, largely neglecting questions of statistical frameworks (frequentist vs. Bayesian) or post-fitting procedures such as inference and prediction. For example, having MCMC samples from a fitted model allows a wide range of inferential and predictive procedures.

Here we introduce a new R package, **glmmTMB**, that estimates GLMs, GLMMs and extensions of GLMMs including zero-inflated and hurdle GLMMs using ML. The ability to fit these types of models quickly and using a single package will make it easier to perform model selection. We focus on zero-inflated counts, but note that there are many other distributions available in **glmmTMB**, including continuous distributions. We demonstrate the package using two examples without going into details of the reasons why a user would want to fit these models. We use an example of salamander abundance to show how to fit and compare zero-inflated and hurdle GLMMs and then how to extract results from a model. We then use variations of the salamander data to compare how the timings of different packages scale with the number of observations and random effect levels. We use a classic example of owl nestling behavior to compare the timing and parameter estimates from **glmmTMB** with other R packages.

Implementation of glmmTMB

The design goal of **glmmTMB** is to extend the flexibility of GLMMs in R while maintaining a familiar interface. To maximize flexibility and speed, **glmmTMB**'s estimation is done using the **TMB** package (Kristensen et al., 2016), but users need not be familiar with **TMB**. We based **glmmTMB**'s interface (e.g., formula syntax) on the **lme4** package — one of the most widely used R packages for fitting GLMMs (Bates et al., 2015). Like **lme4**, **glmmTMB** uses MLE and the Laplace approximation to integrate over random effects; unlike **lme4**, **glmmTMB** does not have the alternative options of doing restricted maximum likelihood (REML) estimation nor using Gauss-Hermite quadrature to integrate over random effects (Bolker et al., 2009; Bolker, 2015). The Laplace approximation may perform poorly when there is little information available on each random effect level (Ogden, 2015). REML may be added to **glmmTMB** in the future. The underlying implementation using **TMB** is a fundamental difference compared to **lme4** and provides **glmmTMB** with a speed advantage when estimating non-Gaussian models (Figures 1 and 2) and gives it greater flexibility in the classes of distributions it can fit (Table 1).

The flexibility of **glmmTMB** enables users to fit and compare many varieties of models with assurance that the log-likelihood values are calculated in a consistent way. Comparing likelihoods of models fit by multiple packages must be done carefully because some packages drop constants from the log-likelihood calculations while others do not.

A **glmmTMB** model has four main components: a conditional model formula, a distribution for the conditional model, a dispersion model formula, and a zero-inflation model formula. Simple GLMs and GLMMs can be fit using the conditional model while leaving the zero-inflation and dispersion

Feature	glmmTMB	glmmADMB	MCMCglmm	brms	INLA	mgcv	gamlss
hurdle models[1]	✓		✓	✓	✓		✓
predictors of zero-inflation	✓		✓	✓			✓
predictors of dispersion	✓			✓			✓
zero-truncated distributions	✓	✓	✓	✓	✓		✓
`nbinom2` distribution	✓	✓		✓	✓	✓[2]	✓
`nbinom1` distribution	✓	✓					✓[2]
`compois` distribution	✓[3]						
Delaporte distribution							✓[2]
Sichel distribution							✓
geometric distribution			✓	✓			✓[2]
PIG distribution							✓
weights	✓[4]	✓		✓	✓	✓	✓
offsets	✓	✓	[5]	✓	✓	✓	
various RE structures	✓[6]		✓	✓	✓	✓	
RE cor across components [7]			✓	✓			
MLE	✓	✓				✓	✓
MCMC samples[8]	✓[9]	✓	✓	✓	✓[10]	✓	
multivariate responses [11]			✓	✓			
GAM [12]				✓		✓	✓

Table 1: Features of packages that are used for modeling zero-inflated count data. This table only contains packages that can at least fit zero-inflated Poisson GLMMs. **lme4** is omitted because it can only estimate zero-inflation when wrapped in an expectation maximization algorithm (Bolker et al., 2013). `nbinom2` and `nbinom1` are negative binomial distributions in which the variance increases quadratically and linearly (respectively) with the mean. `compois` is the Conway-Maxwell-Poisson distribution. PIG is the Poisson inverse Gaussian distribution. Notes: 1 We restrict this to mean a single function call, rather than two separate models. 2 Not available with zero-inflation. 3 Conway-Maxwell-Poisson distribution parameterized by the mean. 4 Weights are often used to reduce the influence of some observations over others, e.g., Gurevitch and Hedges, 1999; additionally, `glmmTMB`'s dispersion formula can be used to model heteroskedasticity. 5 Offsets can be implemented using priors. 6 See `vignette("covstruct")` for details. 7 Some packages allow for correlation across random effects from different components of the model (e.g., conditional and zero-inflation). 8 Here, we mean that MCMC samples can be obtained from an estimated model (i.e., joint samples from the full distribution, conditional on the data) whether it is estimated using MCMC sampling or MLE. In the case of MLE, flat priors are used for MCMC sampling and chains are initiated at the ML estimate. 9 See `vignette("mcmc")` for details. 10 This feature is available and widely used e.g., Chevin et al., 2015, but apparently unsupported by any formal evaluation. 11 We exclude the possibility that the correlation structure of random effects makes a model mathematically equivalent to a multivariate response. 12 Here, we mean generalized additive modeling with automatic smoothness selection. However, a spline can be included in any of these methods using the `bs` or `ns` functions from the **splines** package.

formulas at their default values. The mean of the conditional model is specified using a two-sided formula with the response variable on the left and predictors on the right, potentially including random effects and offsets. This formula uses the same syntax as **lme4**. For example, if salamander counts vary by species (`spp`) and vary randomly by site, then the formula for the dependence of mean count on species could be

```
count ~ spp + (1 | site)
```

The distribution around the mean of the conditional model is specified using the argument `family`. For the types of count data described in the introduction, the distribution will typically be either Poisson or negative binomial. The Conway-Maxwell-Poisson is a less commonly known distribution for count data that is flexible enough to fit both over- and underdispersed data. Following the standard of **glmmTMB** described above, the Conway-Maxwell-Poisson distribution (`family = compois`) is parameterized with the mean (Huang, 2017), which differs from the **COMPoissonReg package**(Sellers et al., 2017). The Poisson, Conway-Maxwell-Poisson, and negative binomial distributions use a log link by default, but other links can be specified as in `family = poisson(link = "identity")`. **glmmTMB** provides two parameterizations of the negative binomial which differ in the dependence of the variance (σ^2) on the mean (μ). For `family = nbinom1`, the variance increases linearly with the mean as $\sigma^2 = \mu(1+\alpha)$, with $\alpha > 0$; for `family = nbinom2`, the variance increases quadratically with the mean as $\sigma^2 = \mu(1+\mu/\theta)$, with $\theta > 0$ (Hardin and Hilbe, 2007). For the Conway-Maxwell-Poisson distribution, there is no closed form equation for the variance (Huang, 2017).

With the default dispersion model (`dispformula = ~ 1`), the dispersion parameter (e.g., α or θ for the negative binomial distribution) is identical for each observation. Alternatively, the dispersion parameter can vary with fixed effects; in this case, the dispersion model uses a log link. The dispersion model can be used to account for heteroskedasticity. For example, if the response is more variable (relative to the mean) as the year progresses, then a model with either negative binomial distribution might use the one-sided formula `dispformula = ~ DOY` where `DOY` is the day of the year. When the same variables are in the conditional and dispersion models, the mean-variance relationship can be manipulated, but this could potentially lead to non-convergence issues. A description of the dispersion

parameter for each distribution can be accessed by typing `?sigma.glmmTMB` in R.

The zero-inflation model describes the probability of observing an extra (i.e., structural) zero that is not generated by the conditional model. Zero-inflation creates an extra point mass of zeros in the response distribution; the overall distribution is a mixture of the conditional model and zero-inflation model (Lambert, 1992; Rhodes, 2015). The zero-inflation probability is bounded between zero and one by using a logit link. For example, if salamanders emerged seasonally at each site, such that a structural zero could occur either because a site was contaminated or because it was visited too early in the season, then the model could include the one-sided formula `ziformula = ~ DOY`. The probability of producing a structural zero can be modeled as equal for all observations with `ziformula = ~ 1`. In **glmmTMB**, it is possible to include random effects in the conditional and zero-inflation models, but not the dispersion model.

Installation of glmmTMB

The package is available from The Comprehensive R Archive Network (CRAN) via the command `install.packages("glmmTMB")`. The current version is 0.2.0. Development versions are available from GitHub and can be installed using **devtools** (Wickham and Chang, 2017). Current details for installing development versions should be accessed on the GitHub page `https://github.com/glmmTMB/glmmTMB`.

Examples and benchmarks

To illustrate how to use **glmmTMB** and to compare it to other packages, we applied it to two data sets that are distributed with **glmmTMB**. Additional code and graphs for these examples can be found in Appendicies A and B.

ABUNDANCE OF SALAMANDERS IN STREAMS

Salamander data We demonstrate how to use features of `glmmTMB` to do model selection and output model results using a data set of the abundance of salamanders (Figure 3). They were observed four times at 23 sites in streams, some of which were impacted by coal mining; multiple species and life stages of salamanders were observed. The data set contains covariates that may affect the habitat suitability of a site and the ability of researchers to capture salamanders that inhabit the site (Price et al., 2016, 2015). Price et al. analyzed the data using a Bayesian model with an ecological and a sampling component; abundance was estimated with a hurdle Poisson model and then observations were modeled as binomial samples from the abundance.

Model fitting and selection We fit GLMMs, zero-inflated GLMMs, and hurdle models to the salamander data with Poisson, Conway-Maxwell-Poisson, and negative binomial distributions on the conditional model. For simplicity, we neglected some possible covariates. As a null model, we assumed that counts varied by species (`spp`) and randomly by site (`site`). We fit models where the mean count additionally depended on mining status (`mined`). Our full zero-inflated GLMMs allowed both the conditional and zero-inflation models to differ between mined and unmined sites. Full zero-inflated and hurdle negative binomial GLMMs were fit using the following commands (respectively) :

```
zinb = glmmTMB(count~spp * mined + (1|site), zi=~spp * mined,
        data=Salamanders, family=nbinom2)
hnb = glmmTMB(count~spp * mined + (1|site), zi=~spp * mined,
        data=Salamanders, family=truncated_nbinom2)
```

As is generally the case for model formulas in R, the `*` indicates an interaction plus main effects. We used Akaike information criteria (AIC) to compare all models via the `AICtab` function from the `bbmle` package (Bolker and Team, 2017). For convenience, `glmmTMB` reports the log-likelihood of unconverged models as `NA` and version 1.0.19 of `bbmle` puts these models at the bottom of AIC tables. The code for fitting these models and doing model selection is presented in Appendix A.

Of the models we considered, the most parsimonious was a Conway-Maxwell-Poisson GLMM that allowed counts to vary with species, mining, and their interaction. It did not include zero-inflation, as is common in abundance models (Warton, 2005). Two zero-inflated negative binomial and one zero-inflated Conway-Maxwell-Poisson model did not converge. Failed convergence is typically caused by trying to estimate parameters for which the data do not contain information. Models that do not converge should not be considered in model comparison. General model convergence issues are discussed in `vignette("troubleshooting",package="glmmTMB")`.

Model summary The summary of simple GLMMs from **glmmTMB** is modeled on the familiar output format of **lme4**. To demonstrate the extra output from zero-inflation and dispersion models, we present the summary from a more complicated model.

```
glmmTMB(count ~ mined + (1|site), zi=~mined , disp=~DOY, Salamanders, family=nbinom2)
```

Following the arguments in the function call above, this model allows the conditional mean to depend on whether or not a site was mined and to vary randomly by site. It allows the number of structural (i.e., extra) zeros to depend on mining. Additionally, it allows the dispersion parameter to depend on the day of the year. This model can be represented by the following set of equations

$$\begin{aligned}
\mu &= E(\texttt{count}|u,\texttt{NSZ}) = \exp(\beta_0 + \beta_{\texttt{minedno}} + u), && (1)\\
u &\sim N\left(0, \sigma_u^2\right), && (2)\\
\sigma^2 &= \mathrm{Var}(\texttt{count}|u,\texttt{NSZ}) = \mu(1+\mu/\theta), && (3)\\
\mathrm{logit}(p) &= \beta_0^{(\texttt{zi})} + \beta_{\texttt{minedno}}^{(\texttt{zi})}, && (4)\\
\log(\theta) &= \beta_0^{(\texttt{disp})} + \beta_{\texttt{DOY}}^{(\texttt{disp})} \cdot \texttt{DOY}, && (5)
\end{aligned}$$

where u is a site specific random effect, NSZ is the event "non-structural zero", $p = 1 - \Pr(\texttt{NSZ})$ is the zero inflation probability, and β's are regression coefficients with subscript denoting covariate/level (with 0 denoting intercept).

```
summary(glmmTMB(count~mined+(1|site), zi=~mined , disp=~DOY, Salamanders, family=nbinom2))
```

```
#>  Family: nbinom2  ( log )
#> Formula:          count ~ mined + (1 | site)
#> Zero inflation:         ~mined
#> Dispersion:             ~DOY
#> Data: Salamanders
#>
#>      AIC      BIC   logLik deviance df.resid
#>     1735     1767     -861     1721      637
#>
#> Random effects:
#>
#> Conditional model:
#>  Groups Name        Variance Std.Dev.
#>  site   (Intercept) 0.134    0.366
#> Number of obs: 644, groups:  site, 23
#>
#> Conditional model:
#>             Estimate Std. Error z value Pr(>|z|)
#> (Intercept)   -0.540      0.376   -1.44     0.15
#> minedno        1.424      0.365    3.90 0.000098 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Zero-inflation model:
#>             Estimate Std. Error z value Pr(>|z|)
#> (Intercept)    0.256      0.487    0.53   0.5988
#> minedno       -2.244      0.745   -3.01   0.0026 **
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Dispersion model:
#>             Estimate Std. Error z value Pr(>|z|)
#> (Intercept)   0.0278     0.3177    0.09     0.93
#> DOY          -0.3947     0.1537   -2.57     0.01 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This summary can be broken down into five sections. The top section is a general overview containing a description of the model specification (`Family`, `Formula`, `Zero inflation`, `Dispersion`,

Data) and resulting information criteria. The information criteria can only be compared to models fitted by packages that, like, **glmmTMB**, compute the full form of the log-likelihood without dropping constant terms. The second section describes the variability of the Random effects. In this model, we only had random effects in the conditional model (equation 1), but random effects from the zero-inflation model (equation 4) could also appear here. The estimated variance 0.134 is σ_u^2 in equation 2. The third section describes the coefficients of the Conditional model (β_0 and $\beta_{\texttt{minedno}}$ in equation 1) including Wald Z statistics and p-values. Apart from the intercept, the estimates are all contrasts as is standard in regression models. This model has a log link as stated in the top line of the summary. The fourth section describes the Zero-inflation model similarly to the Conditional model except that this model has a logit link. The zero-inflation model estimates the probability of an extra zero such that a positive contrast indicates a higher chance of absence (e.g. minedno <0 means fewer absences in sites unaffected by mining); this is the opposite of the conditional model where a positive contrast indicates a higher abundance (e.g., minedno >0 means higher abundances in sites unaffected by mining). The estimates in this section correspond to $\beta_0^{(\texttt{zi})}$ and $\beta_{\texttt{minedno}}^{(\texttt{zi})}$ in equation 4. The last section provides estimated coefficients from the Dispersion model (equation 5), which uses a log link to keep the dispersion parameter θ positive. In contrast, a model with the default (simple) dispersion model would report the single dispersion parameter on the natural (rather than log) scale. To interpret the dispersion parameters of any distribution, see ?sigma.glmmTMB.

The current version of the summary function does not display uncertainty estimates for the random effects nor for single dispersion parameters, but confidence intervals can be calculated using the confint function. See ?confint.glmmTMB for details. All confidence intervals produced by the current version of the confint function are Wald intervals, based on the standard errors calculated using the delta method for the parameter on the scale of the internal parameterization (which varies by family).

Additional model output using the predict and simulate functions from glmmTMB is demonstrated in appendix A (Figures 4, 5, 6, and 7).

Timing comparisons Because **glmmTMB** is the only package that can fit Conway-Maxwell-Poisson GLMMs, it was not possible to do benchmarking with the most parsimonious model. Therefore, for benchmarking, we used the second best model (Δ AIC=0.5) which substituted a negative binomial response for the Conway-Maxwell-Poisson response. We measured the time required to fit this model using multiple packages. We performed three sets of timing benchmarks: (1) on simulated data with the same structure as the original salamander data, (2) on the original data replicated to create more observations per random effect level and the same number of random effect levels, (3) on simulated data with increasing numbers of random effect levels and the same number of observations per random effect level. Benchmarks were run in parallel using parLapply on a high performance computing cluster with 12 cores. This performance should match running on a single core. We used default values for all packages except with **glmmADMB** which required an additional argument (extra.args="-ndi 1000000") to allocate additional memory. By default, **brms** runs four MCMC chains while **MCMCglmm** runs one, which greatly affects their estimation time. However, it would be simple to speed up fitting of **brms** models by running the chains in parallel. For Bayesian methods, the important aspect of timing is sampling efficiency (minimum effective samples per unit time, Bürkner, 2017), but this is not compatible with the MLE methods, so we limit our presentation of the timings of the Bayesian methods.

Benchmarking showed that fitting the negative binomial model to simulated data with the same structure as the original data was, on average, equally fast in glmmTMB and INLA, 14 times slower with glmmADMB, 29 times slower with **lme4**, and 190 times slower with brms. **mgcv** fit the model the fastest, taking 0.03 times as long as **glmmTMB**. **gamlss** took 0.24 times as long as **glmmTMB**. With increasing numbers of observations, the estimation times of all packages appeared to follow power law functions (Figure 1). For simulated data sets with increasing numbers of random effect levels, estimation time increased as a power-law function for all packages except **INLA** which had estimation times that accelerated (Figure 2). The speed of glmmTMB for models with more random effect levels is due to the sparseness handling by TMB. Benchmarking nuances such as memory usage and how timings scale with model complexity could be investigated in more detail in future studies.

BEGGING BEHAVIOR OF OWL NESTLINGS

To further compare R packages for fitting zero-inflated GLMMs, we analyzed counts of begging behavior by owl nestlings. The full analyses can be found in Appendix B. This example previously appeared in Zuur et al. (2009) and Bolker et al. (2013) and was originally published by Roulin and Bersier (2007). We compared the estimates of fixed effects and the amount of time required for fitting the same model in **INLA**, **MCMCglmm**, **glmmADMB**, **mgcv**, and **brms** (Rue et al., 2009; Hadfield, 2010; Skaug et al., 2012; Wood et al., 2016; Bürkner, 2017). For **brms** and **MCMCglmm**, we used the

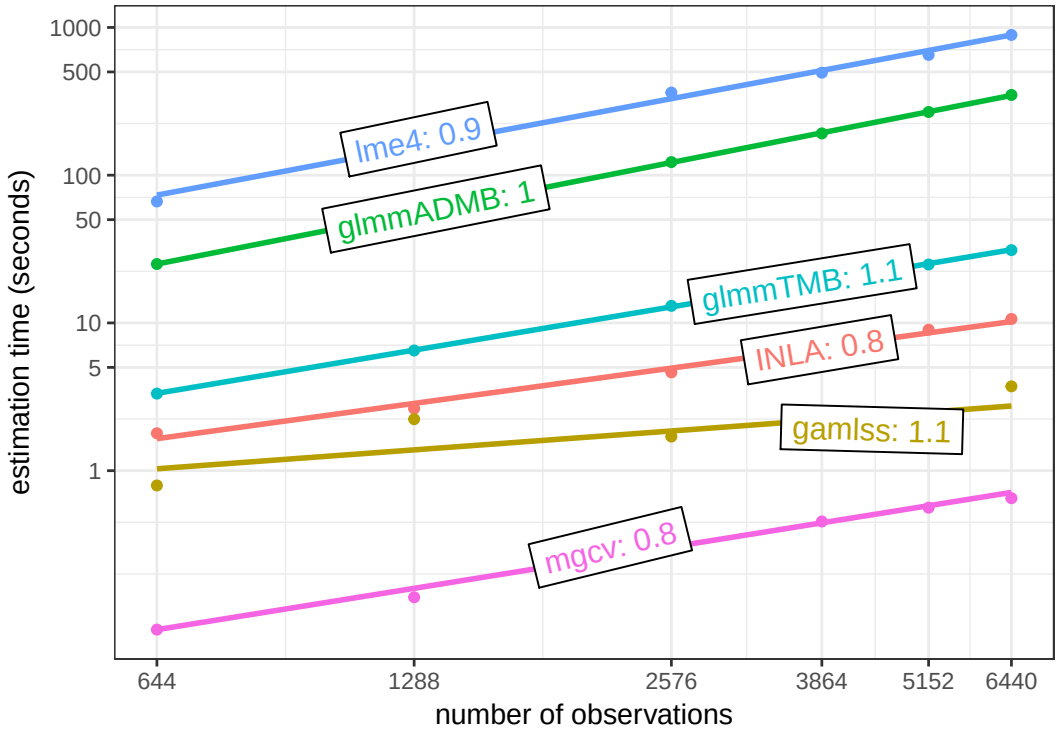

Figure 1: The Salamander data set was replicated by 1, 2, 4, 6, 8, and 10 times to create larger data sets. The time required to fit the same model using functions `glmmadmb`,`glmmTMB`,`inla`, `glmer.nb`, and `gam` was recorded. That model can be represented as `glmmTMB(count ~ spp * mined + (1 | site), Salamanders, family="nbinom2")`. All models had the same number of parameters including random effect levels. Lines represent linear models fit on the log-log scale. With increasing numbers of observations (n), estimation times increased as a power-law function (n^x) with exponents (x) reported next to model names.

default number of iterations, burn-in samples, and thinning. In each package, we fit zero-inflated Poisson models with six fixed effects, one random effect; we also accounted for overdispersion, although sometimes (of necessity) in slightly different ways with different packages (e.g., negative binomial vs. log-normal-Poisson models). We allowed zero-inflation to vary with food treatment and vary randomly with nest. See Appendix B for details of these methods, including code.

Estimates and confidence (or credible) intervals (CI) from **brms**, **mgcv**, `MCMCglmm`, and **INLA** were nearly identical to those of **glmmTMB**, when running the Bayesian models with flat priors (Figures 8 and 9).

Conclusions

We have introduced an R package that can quickly estimate a variety of models including GLMs, GLMMs, zero-inflated GLMMs, and hurdle models. By providing this flexibility in a single package, we reduce the need for researchers to learn multiple packages. Another benefit is that models estimated with a single package can be compared using likelihood-based methods including information criteria. Using information criteria to select a model for the salamander data, we found that (among the models we considered) zero-inflation did not improve the fit; we expect that this will be a common result. While **glmmTMB** allows users to easily fit complicated models, a maximally complex model might not be necessary and might not converge, as we saw here. Other packages have many of the features implemented in **glmmTMB**, but none have the ability to fit Conway-Maxwell-Poisson GLMMs. Overall, **glmmTMB** is a very flexible package for modeling count data with zero-inflated GLMMs while still ranking highly in speed comparisons.

Acknowledgements

Thanks to S Price for providing the data on salamanders and for helpful comments on the manuscript. Thanks to P-C Bürkner, J Hadfield, M Bekker-Nielsen Dunbar, and two anonymous reviewers for helpful comments on the manuscript. Thanks to A Zuur for providing the data on owl nestlings.

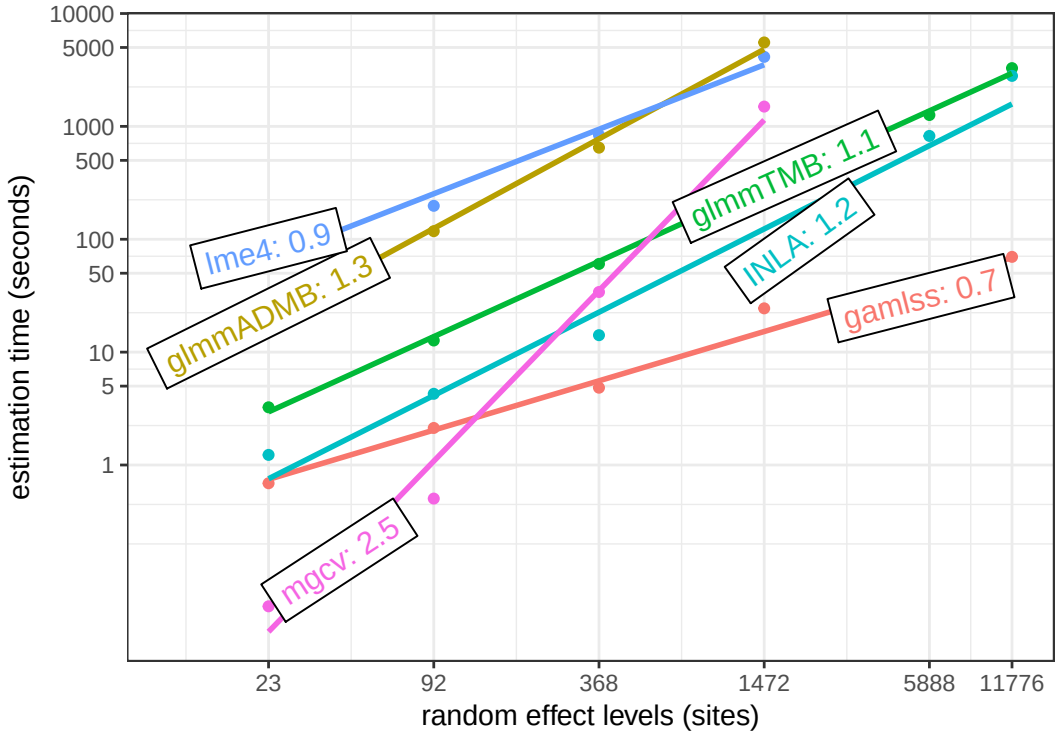

Figure 2: Data sets with increasing numbers of levels of the random effect were simulated based on a negative binomial model fit to the salamander data, `glmmTMB(count ~ spp * mined + (1 | site), Salamanders, family="nbinom2")`. The time required to fit the same model using functions `glmmadmb`, `glmmTMB`, `inla`, `glmer.nb`, and `gam` was recorded. Each simulated data set had the same number of observations per random effect level — the same ratio as in the original data. Lines represent linear models fit on the log-log scale.With increasing numbers of random effect levels (n), estimation times increased as a power-law function (n^x), except for INLA which accelerates. The exponents (x) are reported here as labels over the lines.

Thanks to G Simpson for advice on capabilities of the **mgcv** package. This work was supported by grants from the Swiss National Science Foundation (#IZ320Z0_161670) and from the AD Model Builder Foundation to MEB.

Bibliography

G. D. Barriga and F. Louzada. The Zero-Inflated Conway-Maxwell-Poisson Distribution: Bayesian Inference, Regression Modeling and Influence Diagnostic. *Statistical Methodology*, 21:23–34, 2014. URL https://doi.org/10.1016/j.stamet.2013.11.003. [p378]

D. Bates, M. Mächler, B. M. Bolker, and S. Walker. Fitting Linear Mixed-Effects Models Using Lme4. *Journal of Statistical Software*, 67(1):1–48, 2015. URL https://doi.org/10.18637/jss.v067.i01. [p379]

B. Bolker and R. D. C. Team. *bbmle: Tools for General Maximum Likelihood Estimation*, 2017. URL https://CRAN.R-project.org/package=bbmle. R package version 1.0.20. [p381]

B. M. Bolker. Linear and Generalized Linear Mixed Models. In G. A. Fox, S. Negrete-Yankelevich, and V. J. Sosa, editors, *Ecological Statistics*. Oxford University Press, Oxford, UK, 2015. [p378, 379]

B. M. Bolker, M. E. Brooks, C. J. Clark, S. W. Geange, J. R. Poulsen, M. H. H. Stevens, and J.-S. S. White. Generalized Linear Mixed Models: a Practical Guide for Ecology and Evolution. *Trends in Ecology & Evolution*, 24(3):127–135, 2009. URL https://doi.org/10.1016/j.tree.2008.10.008. [p378, 379]

B. M. Bolker, B. Gardner, M. Maunder, C. W. Berg, M. Brooks, L. Comita, E. Crone, S. Cubaynes, T. Davies, P. de Valpine, J. Ford, O. Gimenez, M. Kéry, E. J. Kim, C. Lennert-Cody, A. Magnusson, S. Martell, J. Nash, A. Nielsen, J. Regetz, H. Skaug, and E. Zipkin. Strategies for Fitting Nonlinear Ecological Models in R, AD Model Builder, and BUGS. *Methods in Ecology and Evolution*, 4(6): 501–512, 2013. URL https://doi.org/10.1111/2041-210x.12044. [p380, 383, 393, 395]

Y. Buckley. Generalized Linear Models. In G. A. Fox, S. Negrete-Yankelevich, and V. J. Sosa, editors, *Ecological Statistics*. Oxford University Press, Oxford, UK, 2015. [p378]

P.-C. Bürkner. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28, 2017. URL https://doi.org/10.18637/jss.v080.i01. [p379, 383]

L. Chevin, M. E. Visser, and J. Tufto. Estimating the variation, autocorrelation, and environmental sensitivity of phenotypic selection. *Evolution*, 69(9):2319–2332, 2015. URL https://doi.org/10.1111/evo.12741. [p380]

M. J. Denwood, M. J. Stear, L. Matthews, S. W. J. Reid, N. Toft, and G. T. Innocent. The Distribution of the Pathogenic Nematode *Nematodirus Battus* in Lambs is Zero-Inflated. *Parasitology*, 135(10): 1225–1235, 2008. URL https://doi.org/10.1017/s0031182008004708. [p378]

D. A. Elston, R. Moss, T. Boulinier, C. Arrowsmith, and X. Lambin. Analysis of aggregation, a worked example: Numbers of ticks on red grouse chicks. *Parasitology*, 122(5):563–569, 2001. URL https://doi.org/10.1017/s0031182001007740. [p378, 394]

B. Grün and F. Leisch. FlexMix Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters. *Journal of Statistical Software*, 28(4):1–35, 2008. URL https://doi.org/10.18637/jss.v028.i04. [p379]

J. Gurevitch and L. V. Hedges. Statistical Issues in Ecological Meta-Analyses. *Ecology*, 80(4):1142–1149, 1999. ISSN 00129658, 19399170. URL https://doi.org/10.2307/177061. [p380]

J. D. Hadfield. MCMC Methods for Multi-Response Generalized Linear Mixed Models: The MCMCglmm R Package. *Journal of Statistical Software*, 33(2):1–22, 2010. URL https://doi.org/10.18637/jss.v033.i02. [p378, 379, 383, 394]

J. W. Hardin and J. M. Hilbe. Generalized Linear Models and Extensions, 2007. [p380]

X. A. Harrison. Using observation-level random effects to model overdispersion in count data in ecology and evolution. *PeerJ*, 2:e616, 2014. ISSN 2167-8359. URL https://doi.org/10.7717/peerj.616. [p378, 379]

X. A. Harrison. A comparison of observation-level random effect and Beta-Binomial models for modelling overdispersion in Binomial data in ecology & evolution. *PeerJ*, 3:e1114, 2015. ISSN 2167-8359. URL https://doi.org/10.7717/peerj.1114. [p378]

A. Huang. Mean-Parametrized Conway–Maxwell–Poisson Regression Models for Dispersed Counts. *Statistical Modelling*, 17(6):1–22, 2017. URL https://doi.org/10.1177/1471082x17697749. [p380]

K. Kristensen, A. Nielsen, C. W. Berg, H. Skaug, and B. Bell. TMB: Automatic Differentiation and Laplace Approximation. *Journal of Statistical Software*, 70(1):1–21, 2016. URL https://doi.org/10.18637/jss.v070.i05. [p379]

V. Lagani, G. Athineou, A. Farcomeni, M. Tsagris, and I. Tsamardinos. Feature selection with the r package mxm: Discovering statistically equivalent feature subsets. *Journal of Statistical Software, Articles*, 80(7):1–25, 2017. URL https://doi.org/10.18637/jss.v080.i07. [p379]

D. Lambert. Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. *Technometrics*, 34(1):1–14, 1992. URL https://doi.org/10.2307/1269547. [p381]

H. J. Lynch, J. T. Thorson, and A. O. Shelton. Dealing with under- and over-dispersed count data in life history, spatial, and community ecology. *Ecology*, 95(11):3173–3180, 2014. URL https://doi.org/10.1890/13-1912.1. [p378]

H. E. Ogden. A sequential reduction method for inference in generalized linear mixed models. *Electronic Journal of Statistics*, 9(1):135–152, 2015. URL https://doi.org/10.1214/15-ejs991. [p379]

R. B. O'Hara and D. J. Kotze. Do Not Log-Transform Count Data. *Methods in Ecology and Evolution*, 1 (2):118–122, 2010. URL https://doi.org/10.1111/j.2041-210x.2010.00021.x. [p378]

S. J. Price, B. L. Muncy, S. J. Bonner, A. N. Drayer, and C. D. Barton. Data From: Effects of Mountaintop Removal Mining and Valley Filling on the Occupancy and Abundance of Stream Salamanders, 2015. URL https://doi.org/10.5061/dryad.5m8f6. [p381, 387]

S. J. Price, B. L. Muncy, S. J. Bonner, A. N. Drayer, and C. D. Barton. Effects of Mountaintop Removal Mining and Valley Filling on the Occupancy and Abundance of Stream Salamanders. *Journal of Applied Ecology*, 53(2):459–468, 2016. URL https://doi.org/10.1111/1365-2664.12585. [p378, 381, 387]

J. R. Rhodes. Mixture Models for Overdispersed Data. In G. A. Fox, S. Negrete-Yankelevich, and V. J. Sosa, editors, *Ecological Statistics*. Oxford University Press, Oxford, UK, 2015. [p378, 381]

A. Roulin and L.-F. Bersier. Nestling Barn Owls Beg More Intensely in the Presence of Their Mother than in the Presence of Their Father. *Animal Behaviour*, 74(4):1099–1106, 2007. URL https://doi.org/10.1016/j.anbehav.2007.01.027. [p378, 383, 393]

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations. *Journal of the royal statistical society: Series B (statistical methodology)*, 71(2):319–392, 2009. URL https://doi.org/10.1111/j.1467-9868.2008.00700.x. [p379, 383]

K. Sellers, T. Lotze, and A. Raim. *COMPoissonReg: Conway-Maxwell Poisson (COM-Poisson) Regression*, 2017. URL https://CRAN.R-project.org/package=COMPoissonReg. R package version 0.4.1. [p380]

K. F. Sellers and G. Shmueli. A Flexible Regression Model for Count Data. *The Annals of Applied Statistics*, 4(2):943–961, 2010. URL https://doi.org/10.1214/09-aoas306. [p378]

G. Shmueli, T. P. Minka, J. B. Kadane, S. Borle, and P. Boatwright. A Useful Distribution for Fitting Discrete Data: Revival of the Conway-Maxwell-Poisson Distribution. *Journal of the Royal Statistical Society C*, 54(1):127–142, 2005. URL https://doi.org/10.1111/j.1467-9876.2005.00474.x. [p378]

H. Skaug, D. Fournier, A. Nielsen, A. Magnusson, and B. M. Bolker. *glmmADMB: Generalized Linear Mixed Models Using AD Model Builder*, 2012. [p379, 383]

M. D. Stasinopoulos, R. A. Rigby, G. Z. Heller, V. Voudouris, and F. De Bastiani. *Flexible Regression and Smoothing: Using GAMLSS in R*. CRC Press, 2017. [p378, 379]

C. Wang, P. R. Torgerson, J. Höglund, and R. Furrer. Zero-inflated hierarchical models for faecal egg counts to assess anthelmintic efficacy. *Veterinary Parasitology*, 235:20 – 28, 2017. ISSN 0304-4017. URL https://doi.org/10.1016/j.vetpar.2016.12.007. [p378]

D. I. Warton. Many Zeros Does Not Mean Zero Inflation: Comparing the Goodness-of-Fit of Parametric Models to Multivariate Abundance Data. *Environmetrics*, 16(3):275–289, 2005. URL https://doi.org/10.1002/env.702. [p381]

H. Wickham and W. Chang. *devtools: Tools to Make Developing R Packages Easier*, 2017. URL https://CRAN.R-project.org/package=devtools. R package version 1.13.4. [p381]

K. Wilson and B. T. Grenfell. Generalized Linear Modelling for Parasitologists. *Parasitology Today*, 13 (1):33–38, 1997. [p378]

S. N. Wood, N. Pya, and B. Säfken. Smoothing Parameter and Model Selection for General Smooth Models. *Journal of the American Statistical Association*, 111(516):1548–1563, 2016. URL https://doi.org/10.1080/01621459.2016.1180986. [p379, 383]

T. W. Yee. *VGAM: Vector Generalized Linear and Additive Models*, 2017. URL https://CRAN.R-project.org/package=VGAM. R package version 1.0-4. [p379]

A. Zeileis, C. Kleiber, and S. Jackman. Regression Models for Count Data in R. *Journal of Statistical Software*, 27(8):1–25, 2008. URL https://doi.org/10.18637/jss.v027.i08. [p379]

A. F. Zuur, E. N. Ieno, N. J. Walker, A. A. Saveliev, and G. M. Smith. *Mixed Effects Models and Extensions in Ecology with R*. Springer-Verlag, 2009. [p383, 393]

Appendix A: Salamander example comparing GLMMs, zero-inflated GLMMs, and hurdle models using glmmTMB

In this appendix, we reanalyze counts of salamanders in streams. Repeated samples of salamanders were taken at 23 sites. Some of the sites were affected by mountain top removal coal mining. The data was originally published in (Price et al., 2016) and was acquired from Dryad (Price et al., 2015). These analyses are intended to be a simple demonstration of how to use some features of the **glmmTMB** package, so we do not attempt to fit all of the models that could be reasonable to try with the covariates that were collected.

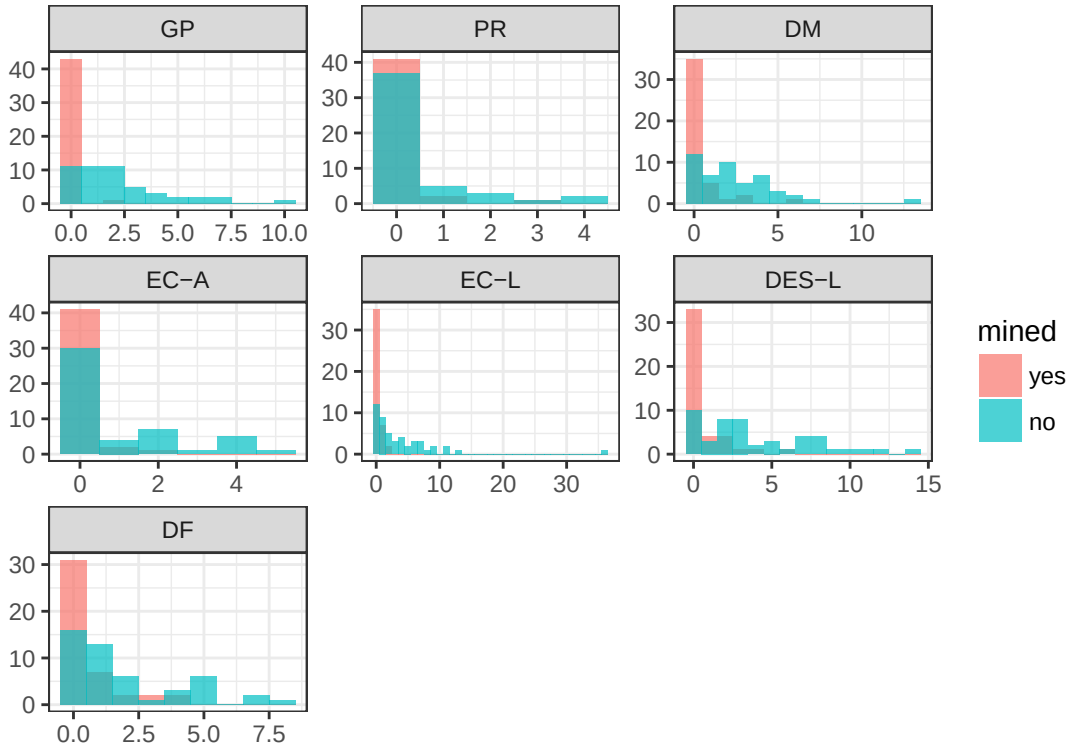

Figure 3: Observed numbers of salamanders. Histograms show count data split into separate panels for each salamander species or life stage. Each panel contains two overlaid histograms in which color represents whether the site was affected by mining.

Poisson models

The syntax for fitting GLMMs with **glmmTMB** is quite similar to using **glmer**. In the first model, the formula, `count ~ spp + (1 | site)`, says that counts depend on species and vary randomly by site. We also pass it the data frame, `Salamanders`, and specify a Poisson distribution using the `family` argument. **glmmTMB** assumes that we want a log link with the Poisson distribution because that's the standard.

```
pm0 = glmmTMB(count~spp + (1|site), Salamanders, family=poisson)
pm1 = glmmTMB(count~spp + mined + (1|site), Salamanders, family=poisson)
pm2 = glmmTMB(count~spp * mined + (1|site), Salamanders, family=poisson)
```

Conway-Maxwell-Poisson models

To fit Conway-Maxwell-Poisson models, we use `family=compois` instead of `poisson`.

```
cmpm0 = glmmTMB(count~spp + (1|site), Salamanders, family=compois)
cmpm1 = glmmTMB(count~spp + mined + (1|site), Salamanders, family=compois)
cmpm2 = glmmTMB(count~spp * mined + (1|site), Salamanders, family=compois)
```

Negative binomial models

```
nbm0 = glmmTMB(count~spp + (1|site), Salamanders, family=nbinom2)
nbm1 = glmmTMB(count~spp + mined + (1|site), Salamanders, family=nbinom2)
nbm2 = glmmTMB(count~spp * mined + (1|site), Salamanders, family=nbinom2)
```

Unlike the Poisson, the negative binomial distribution has a dispersion parameter. If we expected the counts to become more dispersed (relative to the mean) as the year progresses, then we could use the dispersion formula to model how the dispersion changes with the day of the year (`DOY`) using `disp= ~ DOY`.

```
nbdm0 = glmmTMB(count~spp + (1|site), disp=~DOY, Salamanders, family=nbinom2)
nbdm1 = glmmTMB(count~spp + mined + (1|site), disp=~DOY, Salamanders, family=nbinom2)
nbdm2 = glmmTMB(count~spp * mined + (1|site), disp=~DOY, Salamanders, family=nbinom2)
```

Zero-inflated models

To fit zero-inflated models, we use the `ziformula` argument, or **glmmTMB** will also recognize `zi`. This is a formula that describes how the probability of an extra zero (i.e., structural zero) will vary with predictors. In this example, we might assume that absences will at least vary by species (`spp`), so we write `zi= ~ spp`. This formula only has a right side because the left side is always the probability of having a structural zero in the response that was specified in the first formula. The zero-inflation probability is always modeled with a logit link to keep it between 0 and 1.

Zero-inflation can be used with any of the distributions in **glmmTMB**, so we compare the same conditional and zero-inflation models with Poisson, Conway-Maxwell-Poisson, and negative binomial distributions.

Warning messages tell us that `zicmpm3`, `zinbm0`, and `zinbm1` do not converge. The convergence warning refers to `vignette("troubleshooting")`; this vignette will expand as advice for troubleshooting convergence issues develops. It seems that `zicmpm3` is overparameterized, but the problem with `zinbm0` and `zinbm1` is that they only have `mined` in the conditional model and not the zero-inflation model, so that they do not agree well with the data. Plotting the data and thinking carefully about the models should help to avoid convergence issues.

```
zipm0 = glmmTMB(count~spp +(1|site), zi=~spp, Salamanders, family=poisson)
zipm1 = glmmTMB(count~spp + mined +(1|site), zi=~spp, Salamanders, family=poisson)
zipm2 = glmmTMB(count~spp + mined +(1|site), zi=~spp + mined, Salamanders, family=poisson)
zipm3 = glmmTMB(count~spp * mined +(1|site), zi=~spp * mined, Salamanders, family=poisson)

zicmpm0 = glmmTMB(count~spp +(1|site), zi=~spp, Salamanders, family=compois)
zicmpm1 = glmmTMB(count~spp + mined +(1|site), zi=~spp, Salamanders, family=compois)
zicmpm2 = glmmTMB(count~spp + mined +(1|site), zi=~spp + mined, Salamanders, family=compois)
zicmpm3 = glmmTMB(count~spp * mined +(1|site), zi=~spp * mined, Salamanders, family=compois)

#> Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-
#> definite Hessian matrix. See vignette('troubleshooting')

zinbm0 = glmmTMB(count~spp +(1|site), zi=~spp, Salamanders, family=nbinom2)

#> Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-
#> definite Hessian matrix. See vignette('troubleshooting')

zinbm1 = glmmTMB(count~spp + mined +(1|site), zi=~spp, Salamanders, family=nbinom2)

#> Warning in fitTMB(TMBStruc): Model convergence problem; non-positive-
#> definite Hessian matrix. See vignette('troubleshooting')

zinbm2 = glmmTMB(count~spp + mined +(1|site), zi=~spp + mined, Salamanders, family=nbinom2)
zinbm3 = glmmTMB(count~spp * mined +(1|site), zi=~spp * mined, Salamanders, family=nbinom2)

#> Warning in fitTMB(TMBStruc): Model convergence problem; singular
#> convergence (7). See vignette('troubleshooting')
```

Hurdle models

We can also fit hurdle models in a single model by using a truncated distribution for the conditional model and adding zero-inflation.

```
hpm0 = glmmTMB(count~spp + (1|site), zi=~spp, Salamanders, family=truncated_poisson)
hpm1 = glmmTMB(count~spp + mined + (1|site), zi=~spp + mined, Salamanders,
               family=truncated_poisson)
hpm2 = glmmTMB(count~spp * mined + (1|site), zi=~spp + mined, Salamanders,
               family=truncated_poisson)
hnbm0 = glmmTMB(count~spp + (1|site), zi=~spp, Salamanders, family=truncated_nbinom2)
hnbm1 = glmmTMB(count~spp + mined + (1|site), zi=~spp + mined, Salamanders,
               family=truncated_nbinom2)
hnbm2 = glmmTMB(count~spp * mined + (1|site), zi=~spp + mined, Salamanders,
               family=truncated_nbinom2)
```

Model comparison using AIC

We can use `AICtab` to compare all the GLMMs, including zero-inflated and hurdle models. Here, to save space, we only output the AICtable for the top four and bottom four models. The most parsimonious model has a Conway-Maxwell-Poisson distribution with effects of species, mining, and their interaction.

```
AICtab(pm0, pm1, pm2,
       cmpm0, cmpm1, cmpm2,
       nbm0, nbm1, nbm2,
       nbdm0, nbdm1, nbdm2,
       zipm0, zipm1, zipm2, zipm3,
       zicmpm0, zicmpm1, zicmpm2, zicmpm3,
       zinbm0, zinbm1, zinbm2, zinbm3,
       hpm0, hpm1, hpm2,
       hnbm0, hnbm1, hnbm2)
```

The top of the table

model	df	dAIC
cmpm2	16	0.00
nbm2	16	0.48
nbdm2	17	1.99
zicmpm2	18	2.09
zinbm3	30	6.80

and the bottom

model	df	dAIC
hpm0	15	314
pm0	8	330
zicmpm3	30	NA
zinbm0	16	NA
zinbm1	17	NA

The log-likelihood of the unconverged models is reported as `NA` so that these models appear at the end of the AIC table. The negative log-likelihood could be extracted with `zinbm1$fit$objective` if it was needed.

Plotting model results

There are many decisions to make about marginalizing over or conditioning on the random effects. See discussion at https://cran.r-project.org/web/packages/merTools/vignettes/Using_predictInterval.html.

For demonstration purposes, we plot results from the top zero-inflated model `zinbm3`.

Quick and dirty plot

It's easiest to see the pattern by using the `predict` function. To avoid marginalizing over or conditioning on random effects, we can refit the best model without the random effect of site; however, this is not ideal because it ignores the correlation within sites. We present a more rigorous version next.

The `predict` function has a parameter `zitype` that specifies whether you want predictions from the conditional model, the zero-inflation model, or the expected response that combines both parts of the model.

```
zinbm3FE = glmmTMB(count~spp * mined, zi=~spp * mined, Salamanders, family=nbinom2)
newdata0 = newdata = unique(Salamanders[,c("mined","spp")])
temp = predict(zinbm3FE, newdata, se.fit=TRUE, zitype="response")
newdata$predFE = temp$fit
newdata$predFE.min = temp$fit-1.96*temp$se.fit
newdata$predFE.max = temp$fit+1.96*temp$se.fit

real=ddply(Salamanders, ~site+spp+mined, summarize, m=mean(count))

ggplot(newdata, aes(spp, predFE, colour=mined))+geom_point()+
```

```
geom_errorbar(aes(ymin=predFE.min, ymax=predFE.max))+
geom_point(data=real, aes(x=spp, y=m) )+
ylab("Average abundance \n including presences and absences")+
xlab("Species")
```

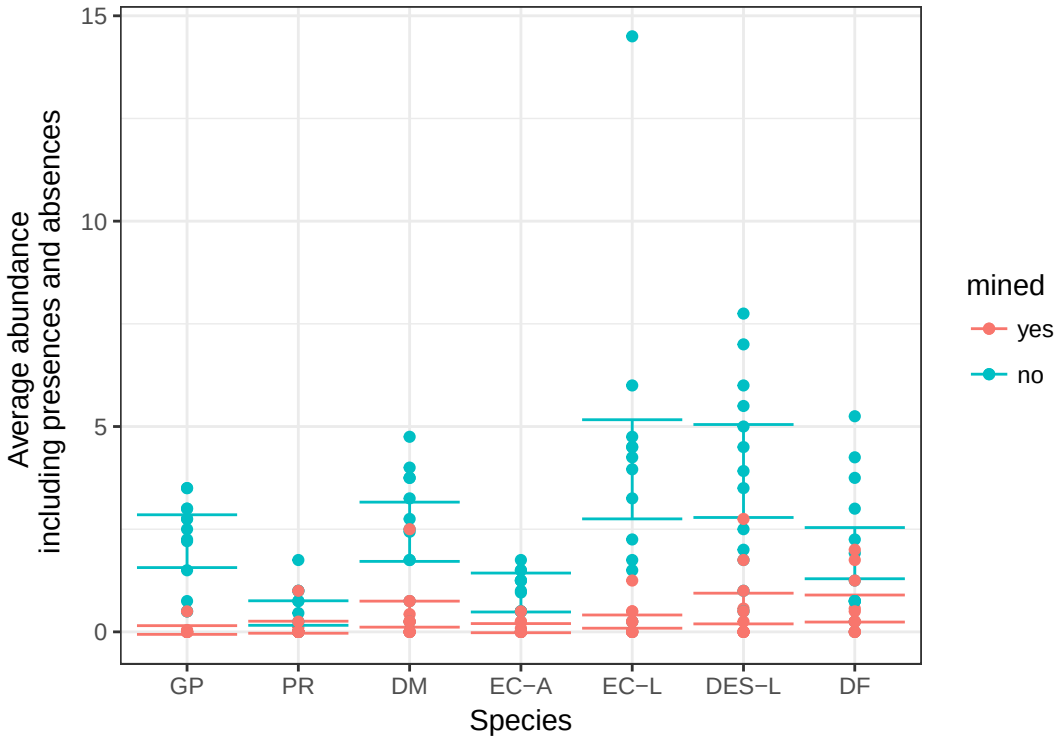

Figure 4: Estimated abundance ignoring correlation. Points represent site-specific average counts. Error bars represent the 95% Wald-type confidence intervals for the predicted average count.

Alternative prediction method

We can predict at the population mode, by setting the random effects to zero.

```
X.cond = model.matrix(lme4::nobars(formula(zinbm3)[-2]), newdata0)
beta.cond = fixef(zinbm3)$cond
pred.cond = X.cond %*% beta.cond

ziformula = zinbm3$modelInfo$allForm$ziformula #$
X.zi = model.matrix(lme4::nobars(ziformula), newdata0)
beta.zi = fixef(zinbm3)$zi
pred.zi = X.zi %*% beta.zi
```

These are estimates of the linear predictors (i.e., predictions on the link scale: logit(prob) and log(cond)), not the predictions themselves. The easiest thing to do for the point estimates of the unconditional count (ucount) is to transform to the response scale and multiply:

```
pred.ucount = exp(pred.cond)*(1-plogis(pred.zi))
```

For the standard errors/confidence intervals, we could use posterior predictive simulations (i.e., draw multivariate normal samples from the parameter for the fixed effects). This conditions on/ignores uncertainty in the random-effect parameters.

```
library(MASS)
set.seed(101)
pred.condpar.psim = mvrnorm(1000,mu=beta.cond,Sigma=vcov(zinbm3)$cond)
pred.cond.psim = X.cond %*% t(pred.condpar.psim)
pred.zipar.psim = mvrnorm(1000,mu=beta.zi,Sigma=vcov(zinbm3)$zi)
pred.zi.psim = X.zi %*% t(pred.zipar.psim)
pred.ucount.psim = exp(pred.cond.psim)*(1-plogis(pred.zi.psim))
```

```
ci.ucount = t(apply(pred.ucount.psim,1,quantile,c(0.025,0.975)))
ci.ucount = data.frame(ci.ucount)
names(ci.ucount) = c("ucount.low","ucount.high")
pred.ucount = data.frame(newdata0, pred.ucount, ci.ucount)
```

These predicted counts should be close to the median counts, so we plot them together to compare.

```
real.count = ddply(Salamanders, ~spp+mined, summarize, m=median(count), mu=mean(count))
ggplot(pred.ucount, aes(x=spp, y=pred.ucount, colour=mined))+geom_point(shape=1, size=2)+
  geom_errorbar(aes(ymin=ucount.low, ymax=ucount.high))+
  geom_point(data=real.count,  aes(x=spp, y=m, colour=mined), shape=0, size=2)+
  geom_point(data=real.count,  aes(x=spp, y=mu, colour=mined), shape=5, size=2)+
  ylab("Abundance \n including presences and absences")+
  xlab("Species")
```

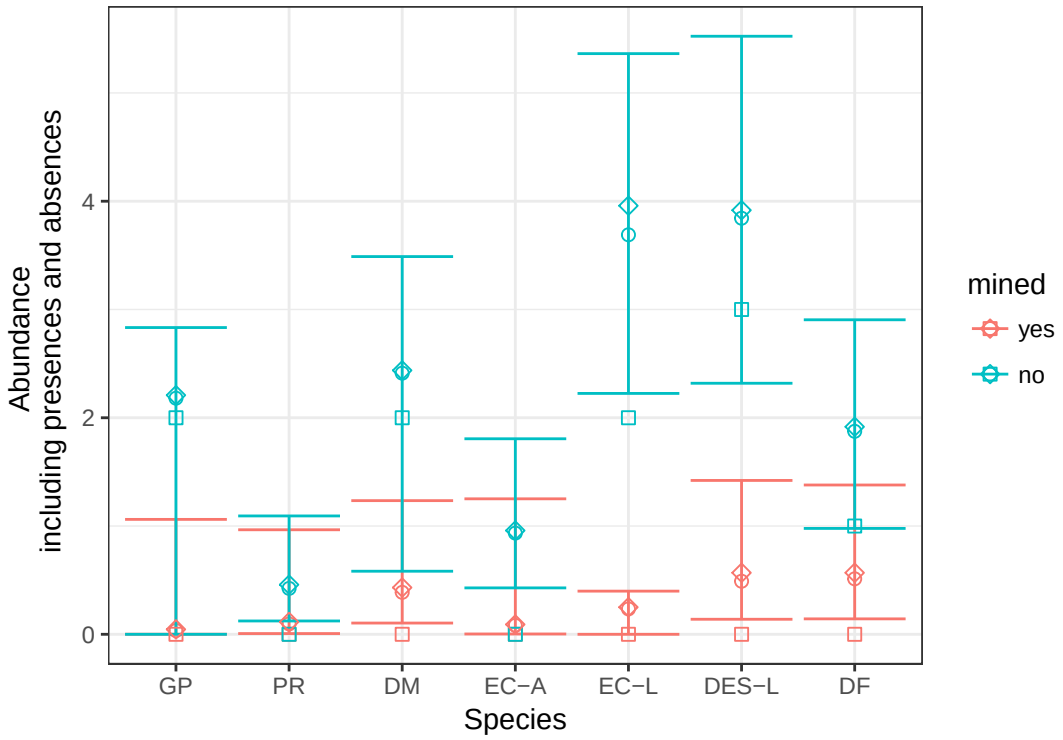

Figure 5: Estimated abundance at mode. Circles represent predicted unconditional counts at the mode (i.e., site effect = 0) and error bars represent the 95% confidence intervals for that mode. Squares represent the observed median and diamonds represent observed means calculated across samples and sites. In this highly skewed data, the mode is closer to the mean than the median.

Simulating from a fitted model

We could also examine the distribution of simulated values from the best fitted model. For this we use the function `simulate.glmmTMB`. This function works for zero-inflated and hurdle models as well as less complex models.

```
sims=simulate(nbm2, seed = 1, nsim = 1000)
```

This function returns a list of vectors. The list has one element for each simulation (`nsim`) and the vectors are the same shape as our response variable.

```
simdatlist=lapply(sims, function(count){
  cbind(count, Salamanders[,c('site', 'mined', 'spp')])
})
simdatsums=lapply(simdatlist, function(x){
  ddply(x, ~spp+mined, summarize,
        absence=mean(count==0),
        mu=mean(count))
```

```
})
ssd=do.call(rbind, simdatsums)
```

Then we can plot them with the observations summarized in the same way.

```
real = ddply(Salamanders, ~spp+mined, summarize,
             absence=mean(count==0),
             mu=mean(count))
ggplot(ssd, aes(x=absence, color=mined))+
  geom_density(adjust=4)+
  facet_wrap(~spp)+
  geom_point(data=real, aes(x=absence, y=1, color=mined), size=2)+
  xlab("Probability that salamanders are not observed")+ylab(NULL)
```

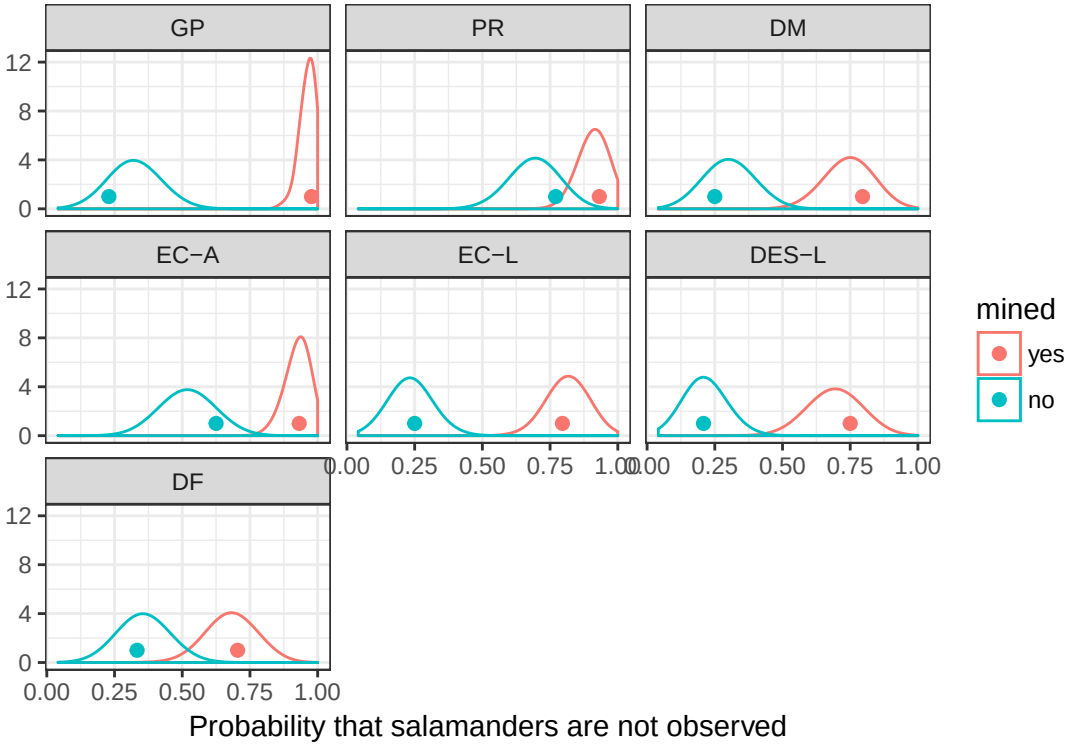

Figure 6: Simulated zero counts. Each panel represents a different species or life stage of a species. Densities are values from 1000 data sets simulated from our best fit model. Points represent the observed data.

We can see that this model does a good job of capturing the observed zero counts.

```
ggplot(ssd, aes(x=mu, color=mined))+
  geom_density(adjust=4)+
  facet_wrap(~spp)+
  geom_point(data=real, aes(x=mu, y=.5, color=mined), size=2)+
  xlab("Abundance including presences and absences")+ylab(NULL)
```

Appendix B: Compare zero-inflated mixed models across R packages

In this appendix, we analyze counts of begging behavior by owl nestlings. This example previously appeared in similar forms (Zuur et al., 2009) and (Bolker et al., 2013); the data were originally published by (Roulin and Bersier, 2007). The response variable is the number of calls from chicks (`NCalls`) in a nest. Two changes from the example published in (Bolker et al., 2013) are that (1) we use an observation-level random effect to account for overdispersion; (2) instead of assuming that the number of calls is strictly proportional to brood size (i.e., using an offset of log(brood size)), we fit the model with log(brood size) as a predictor, equivalent to assuming that calls $\propto$ (brood size)$^\gamma$, with γ not necessarily equal to 1.

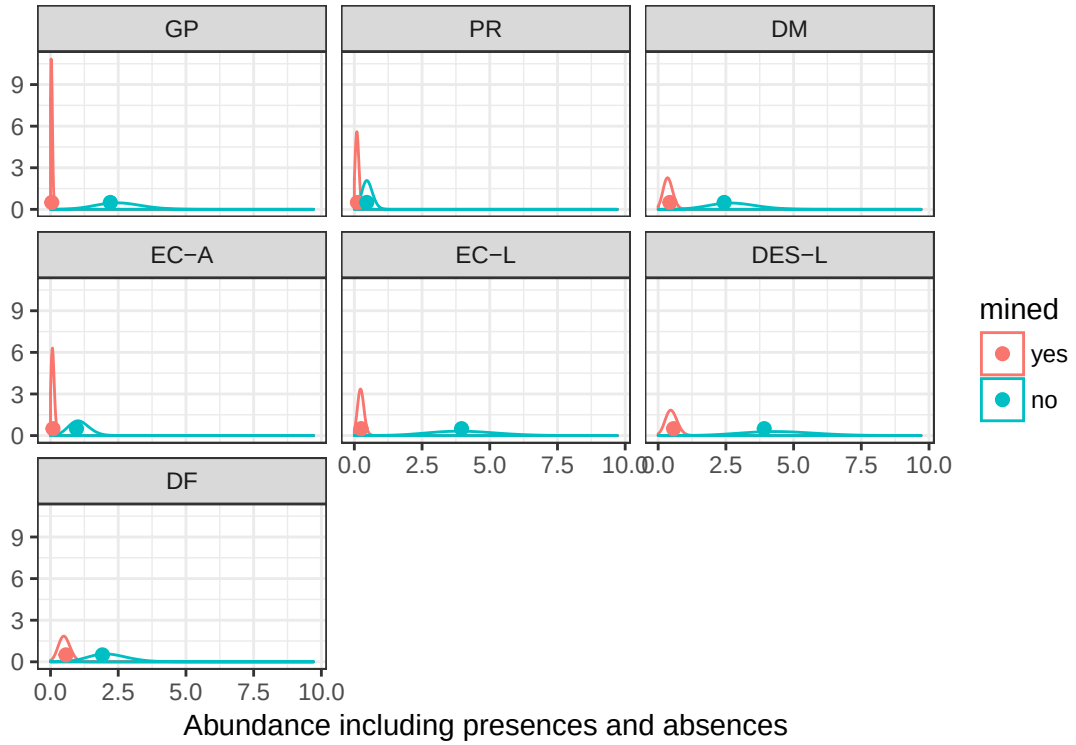

Figure 7: Simulated unconditional abundances. Each panel represents a different species or life stage of a species. Densities are values from 1000 data sets simulated from our best fit model. Points represent the observed data.

Since nests were repeatedly measured, `Nest` is included as a random effect; observation-level random effects are incorporated to allow for overdispersion (Elston et al., 2001; Hadfield, 2010). Covariates of interest include the sex of the parent visiting the nest (`SexParent`), whether the chicks were satiated or not (`FoodTreatment`), and the timing of the parent's arrival (`ArrivalTime`).

Preliminaries

Load packages

```
library(glmmTMB)
library(glmmADMB)
library(MCMCglmm)
library(brms)
library(INLA)
library(mgcv)
library(broom) #for tidy devtools::install_github("bbolker/broom")
library(plyr)
library(dplyr) #tidyverse
library(ggplot2); theme_set(theme_bw())
library(ggstance)#for position_dodgev
```

Data organization and helper functions (hidden)

```
data(Owls,package="glmmTMB")
Owls = plyr::rename(Owls, c(SiblingNegotiation="NCalls"))
Owls = transform(Owls,
                 ArrivalTime=scale(ArrivalTime, center=TRUE, scale=FALSE),
                 obs=factor(seq(nrow(Owls))))
```

Constant zero-inflation

Here we fit the model with zero-inflation assumed to be constant across the data set, i.e., zero-inflation is independent of the predictor variables.

glmmTMB

```
fixef1 = NCalls~(FoodTreatment + ArrivalTime) * SexParent + logBroodSize
form1 =  update(fixef1, . ~ . + (1|Nest)+(1|obs))
time.tmb = tfun(m1.tmb <<- glmmTMB(form1,
                                   ziformula=~1, data = Owls,
                                   family=poisson))
```

glmmADMB

With the additions to the model (`logBroodSize` as a covariate and observation-level random effects), we were unsuccessful in fitting the model with **glmmADMB**. Some variants (e.g., with observation-level random effects, but with `logBroodSize` as an offset) were possible by modifying control settings (i.e. `admb.opts=admbControl(shess=FALSE,noinit=FALSE)`), but even when successful these fits were very slow (>10 minutes).

MCMCglmm

Code for this example was modified from Bolker et al. (2013); a more complete description appears in the supplementary material for that paper.

```
fixef2 = NCalls~trait-1+ # intercept terms for both count and binary terms
    # other fixed-effect terms only apply to count term
    at.level(trait,1):logBroodSize+
    at.level(trait,1):((FoodTreatment+ArrivalTime)*SexParent)
nfix = 8
# residual variances independent for count and binary terms;
#    fixed to 1 for binary term
# random-effects variances independent for count and binary terms;
#    fixed very small (1e-6) for binary term
prior_overdisp  = list(R=list(V=diag(c(1,1)),nu=0.002,fix=2),
                       G=list(list(V=diag(c(1,1e-6)),nu=0.002,fix=2)))
prior_overdisp_broodoff = c(prior_overdisp,
                            list(B=list(mu=rep(0,nfix),
                                        V=diag(rep(1e4,nfix)))))
set.seed(101)
time.MCMCglmm = tfun(m1.MCMCglmm <<- MCMCglmm(fixef2,
                                   rcov=~idh(trait):units,
                                   random=~idh(trait):Nest,
                                   prior=prior_overdisp_broodoff,
                                   data=Owls,
                                   nitt=103000,
                                   thin=100,
                                   family="zipoisson",
                                   verbose=FALSE))
```

We adjusted the number of samples until the effective sample size of the most poorly sampled parameter (the zero-inflation parameter, in this case) was greater than 500; with 100,000 samples after a burn-in of 3,000, we achieved a minimum effective sample size of 568 (for the zero-inflation parameter).

brms

```
time.brms = tfun(m1.brms <<- brm(form1, data = Owls,
                                 iter=1000,
                                 family="zero_inflated_poisson"))

#> Compiling the C++ model
```

```
#> Start sampling
```

One of the known advantages of Hamiltonian Monte Carlo, as implemented in Stan (on which **brms** is built), is that it achieves high effective sample size per MCMC step; we were able to cut down the number of samples considerably from the default of 2000 and still achieve a minimum effective sample size of approximately 500 (the minimum effective sample size achieved was 460).

We resample to estimate the time required for sampling only (i.e., not including model compilation

```
time.brms2 = tfun(m1.brms2 <<- update(m1.brms))
```

```
#> Start sampling
```

INLA

```
form2 = update(fixef1, . ~ . + f(Nest, model="iid") + f(obs, model="iid"))
time.inla = tfun(m1.inla <<-
       inla(form2,
            family= "zeroinflatedpoisson1",
            data=Owls))
```

mgcv

To the best of our knowledge, `mgcv::gam` is currently unable to fit models with the combination of zero-inflation and overdispersion (the `ziplss()` famiy that handles zero-inflation handles only a ZIP case rather than zero-inflated negative binomial or other extensions, and observation-level random effects cannot be fit with `gam`'s random-effect approach). To address this issue, we fitted a GAM with zero-inflated Poisson responses and used a quasi-likelihood approach: i.e., estimating overdispersion as [sum of squared Pearson residuals/residual degrees of freedom] and inflating the parameter standard errors by the square root of the overdispersion.

```
form3 = update(fixef1, . ~ . + s(Nest, bs="re"))
time.mgcv = tfun(m1.gam <<- gam(list(form3, ~ 1),
                                data = Owls,
                                family = ziplss(), method = "REML"))
```

Comparing the results

Timings:

	time (sec)
mgcv	0.2
glmmTMB	2.5
INLA	4.2
MCMCglmm	53.0
brms (no compilation)	55.2
brms (with compilation)	115.6

The deterministic methods (`gam`, `glmmTMB` and `inla`) were all fast; `gam` was fastest, because we fitted a simpler model (see above). The stochastic methods (`MCMCglmm` and `brm`) were about an order of magnitude slower.

Because we ran **brms** with flat priors, the estimates are very close to the ML estimates of **glmmTMB**. The most different results are from `gam`, because we used a quasi-likelihood model with a slightly different implicit variance scaling.

Complex zero-inflation

Here we fit the model with zero-inflation depending on some of the predictor variables. We can no longer use **glmmADMB** or **INLA** (**INLA** allows the zero-inflation probabilities to depend on covariates in hurdle models — "type 0" in the INLA documentation — but not for zero-inflated models).

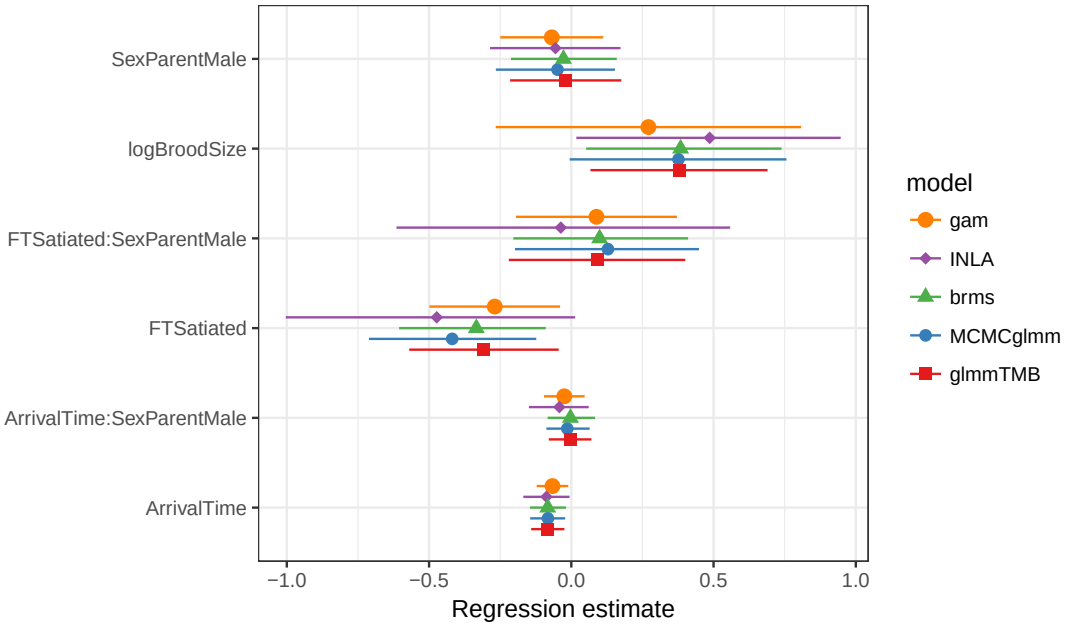

Figure 8: Estimated fixed-effect coefficients: Estimates are from similar zero-inflated Poisson models fit using functions `glmmTMB`, `MCMCglmm`, `brm`, `inla`, and `gam`.

glmmTMB

```
form0 = update(fixef1, . ~ . + (1|Nest))
ziform = ~FoodTreatment+(1|Nest)
time.tmb_czi = tfun(m1.tmb_czi <<- glmmTMB(form0,
                                           ziformula=ziform,
                                           data = Owls, family=nbinom2))
```

Attempting to fit the **glmmTMB** model with a log Normal-Poisson model (i.e., a Poisson model with observation-level random effects) and covariate-dependent zero-inflation led to convergence failure, so we substituted a similar model (a negative-binomial model without observation-level random effects).

MCMCglmm

```
fixef3 = NCalls~trait-1+ # intercept terms for both count and binary terms
    # fixed-effect terms for count term
    at.level(trait,1):logBroodSize+
    at.level(trait,1):((FoodTreatment+ArrivalTime)*SexParent)+
    # fixed-effect terms for binary term
    at.level(trait,2):FoodTreatment
nfix = 9
# residual variances independent for count and binary terms;
#    fixed to 1 for binary term
# random-effects variances now allow estimated variance for binary term
#   as well
prior_overdisp_czi  = list(R=list(V=diag(c(1,1)),nu=0.002,fix=2),
                       G=list(list(V=diag(c(1,1)),nu=0.002)))
prior_overdisp_broodoff_czi = c(prior_overdisp_czi,
                                list(B=list(mu=rep(0,nfix),
                                            V=diag(rep(1e4,nfix)))))
set.seed(101)
time.mcmc_czi=tfun(m1.mcmc_czi <<- MCMCglmm(fixef3,
                                    rcov=~idh(trait):units,
                                    random=~idh(trait):Nest,
                                    prior=prior_overdisp_broodoff_czi,
                                    data=Owls,
                                    nitt=153000,
```

```
                                  family="zipoisson",
                                  verbose=FALSE))
## warning message suppressed ...
```

Minimum effective sample size: 884

brms

```
time.brms_czi = tfun(m1.brms_czi <<- brm(brmsformula(form1, zi=ziform),
                                 data = Owls,
                                 family="zero_inflated_poisson"))

#> Compiling the C++ model

#> Start sampling

time.brms_czi2 = tfun(m1.brms_czi2 <<- update(m1.brms_czi))

#> Start sampling
```

mgcv

```
time.mgcv_czi = tfun(m1.gam_czi <<- gam(list(form3,
                                             ~FoodTreatment+ s(Nest, bs = "re")),
                                    data = Owls, family = ziplss(), method = "REML"))
```

Comparison

Timings:

	time (sec)
mgcv	0.3
TMB	3.9
brms (no compilation)	53.1
MCMCglmm	88.5
brms (with compilation)	174.4

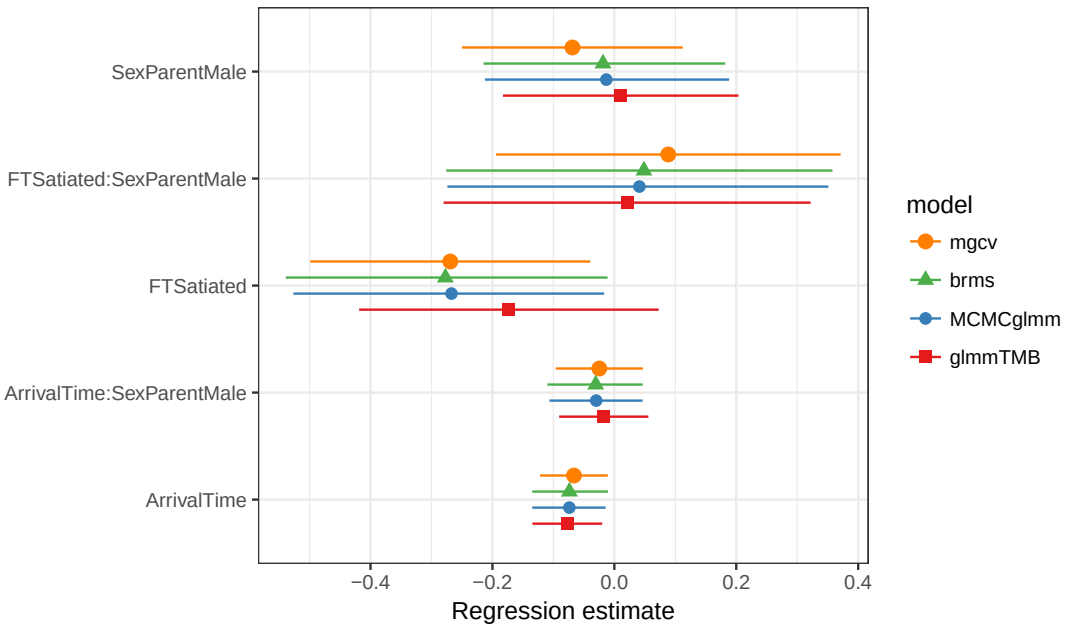

Figure 9: Estimated fixed-effect coefficients: Estimates are from the same zero-inflated Poisson model with predictors on zero-inflation fit using functions `glmmTMB`, `MCMCglmm`, `brm`, and `gam`.

Package versions:

```
#>     brms glmmADMB  glmmTMB     INLA MCMCglmm     mgcv
#>   1.10.2    0.8.5    0.2.0  17.6.20     2.25   1.8.20
```

Mollie E. Brooks
National Institute of Aquatic Resources, Technical University of Denmark
Building 201, 2800 Lyngby
Denmark
Department of Evolutionary Biology and Environmental Studies, University of Zurich
Winterthurerstrasse 190, 8057 Zurich
Switzerland
orcid.org/0000-0001-6963-8326
MollieEBrooks@gmail.com

Kasper Kristensen
National Institute of Aquatic Resources, Technical University of Denmark
Building 201, 2800 Lyngby
Denmark
kaskr@dtu.dk

Koen J. van Benthem
Department of Evolutionary Biology and Environmental Studies, University of Zurich
Winterthurerstrasse 190, 8057 Zurich
Switzerland
koenvanbenthem@gmail.com

Arni Magnusson
International Council for the Exploration of the Sea
H.C. Andersens Boulevard 44-46, 1553 Copenhagen
Denmark
arni.magnusson@ices.dk

Casper W. Berg
National Institute of Aquatic Resources, Technical University of Denmark
Building 201, 2800 Lyngby
Denmark
cbe@aqua.dtu.dk

Anders Nielsen
National Institute of Aquatic Resources, Technical University of Denmark
Building 201, 2800 Lyngby
Denmark
an@aqua.dtu.dk

Hans J. Skaug
Department of Mathematics, University of Bergen
P.O. Box 7803, 5020 Bergen
Norway
Hans.Skaug@uib.no

Martin Mächler
Seminar für Statistik, ETH Zurich
8092 Zurich
Switzerland
orcid.org/0000-0002-8685-9910
Maechler@stat.math.ethz.ch

Benjamin M. Bolker
Department of Mathematics and Statistics, McMaster University
1280 Main St W, L8S4L8 Hamilton, Ontario
Canada
Department of Biology, McMaster University
1280 Main St W, L8S4L8 Hamilton, Ontario

Canada
orcid.org/0000-0002-2127-0443
Bolker@mcmaster.ca