GLMMs in action: gene-by-environment interaction in total fruit production of wild populations of *Arabidopsis thaliana*
Revised version, part 2

Benjamin M. Bolker* Mollie E. Brooks[2] Connie J. Clark[3]
Shane W. Geange[4] John R. Poulsen[3] M. Henry H. Stevens[5]
Jada-Simone S. White[6]

September 14, 2011

1 Introduction

This is revised supplementary material for (Bolker et al., 2009). As described in part 1 of this document, we analyze variation in nutrient availability and herbivory among genotypes and populations of mouse-ear cress (*Arabidopsis thaliana*: Banta et al. (2010)) [1]. Part 1 describes the data and the context of the analysis and gives a detailed example of exploratory graphical analysis, model fitting, model reduction to remove overfitted components, and inference on the fitted model(s). In part 1, we focus on the `glmer` function from the `lme4` package, which is in some ways the most developed package for fitting GLMMs in R.

*[1]Departments of Mathematics & Statistics and Biology, McMaster University, Hamilton, Ontario, Canada L8S 4K1; [2] Department of Biology, University of Florida, PO Box 118525, Gainesville FL 32611-8525; [3] Nicholas School of the Environment, Duke University, PO Box 90328, Durham, NC 27708; [4]School of Biological Sciences, Victoria University of Wellington PO Box 600, Wellington 6140, New Zealand; [5]Department of Botany, Miami University, Oxford OH, 45056; [6]Department of Biological Sciences, California State University, Chico, Chico, CA 9529-515. *Corresponding author*: Bolker, B. M. (`bolker@mcmaster.ca`)

[1]Data and context are provided courtesy of J. Banta and M. Pigliucci, State University of New York (Stony Brook).

In this part, we extend the analysis to compare the procedure and results from `glmer` with other approaches (`MCMCglmm`, `MASS::glmmPQL`, `glmmADMB`, and `lmer` in the `lme4` package).

Load packages:

```
> ## install.packages("coefplot2",repos="http://r-forge.r-project.org")
> library(coefplot2)  ## for coefplot2
> library(reshape)
> library(plyr)
> library(ggplot2)
> library(emdbook)   ## for qchibarsq
> library(bbmle) ## for AICtab
> library(lme4)
> library(MCMCglmm)
> source("glmm_funs.R")
```

Load results from part 1:

```
> load("Banta_part1.RData")
```

[2]

Later we will also load the `glmmADMB` package, but there are conflicts between some of its methods and `lme4`, so we will wait to load it until we really need it.

2 MCMCglmm

Now we'll use the `MCMCglmm` package to fit the model. The main advantages of `MCMCglmm` are that it (1) uses Bayesian numerical integration methods, which are at present one of the most reliable ways to get confidence intervals on variance components and with small data sets, and (2) is quite flexible (it allows estimation of zero-inflated distributions, multivariate data, setting of priors, etc.). Its disadvantages are that it is a bit slower than `glmer` and that, as a side effect of its flexibility, its syntax is slightly more complicated.

[2] We used R version 2.13.1 (2011-07-08) and package versions:

bbmle	coda	coefplot2	ggplot2	glmmADMB	lme4
1.0.3	0.14-4	0.1.1	0.8.9	0.6.3	0.999375-41
MCMCglmm	plyr	reshape	RLRsim		
2.14	1.6	0.8.4	2.0-10		

The `coefplot2` and `glmmADMB` packages must be installed from `http://r-forge.r-project.org`; all others are on CRAN.

```
> library(MCMCglmm)
```

The random effects in an `MCMCglmm` fit are specified as `v(X):g` where `g` is the grouping factor (population or genotype in our case); `X` is the set of fixed effects that vary across the levels of the grouping factor (`1`, `amd`, `nutrient`, etc.); and `v` specifies the structure of the variance-covariance matrix. Using `us` would fit an unstructured (i.e. fully flexible) variance-covariance matrix, as is the default in `glmer`. This failed with the error `ill-conditioned G/R structure: use proper priors if you haven't or rescale data if you have`, so we instead used the `idh` specification which makes the variation in each effect independent.

We fit three models — one with variation in nutrient, clipping, and interaction effects at both levels; one with variation in clipping effects at the population level and intercept variation at the genotype level; and one with intercept-only variation at population and genotype levels (i.e. equivalent to our final model, `mp4`, in Part 1).

```
> ## use independent effects (idh); full covariance matrix (us) fails
> mcmc1 <- MCMCglmm(total.fruits ~ nutrient*amd +
          rack + status,
          random=~idh(amd*nutrient):popu+idh(amd*nutrient):gen,
                   data=dat.tf, family="poisson",verbose=FALSE)
> mcmc2 <- MCMCglmm(total.fruits ~ nutrient*amd +
          rack + status,
          random=~idh(amd):popu+idh(1):gen,
                   data=dat.tf, family="poisson",verbose=FALSE)
> mcmc3 <- MCMCglmm(total.fruits ~ nutrient*amd +
          rack + status,
          random=~idh(1):popu+idh(1):gen,
                   data=dat.tf, family="poisson",verbose=FALSE)
```

Trace plots are the first graphical diagnostic for MCMC-based estimation. For the full model, the trace plots for the fixed effects look as they should, with no evidence of trends or slowly varying temporal patterns (Figure 1.

```
> print(xyplot(as.mcmc(mcmc1$Sol),layout=c(3,3)))
```

However, the variance parameters don't look good (Figure 2) — many of the variance parameters are getting stuck at zero for long periods of time, or taking brief excursions to extreme values, or both.

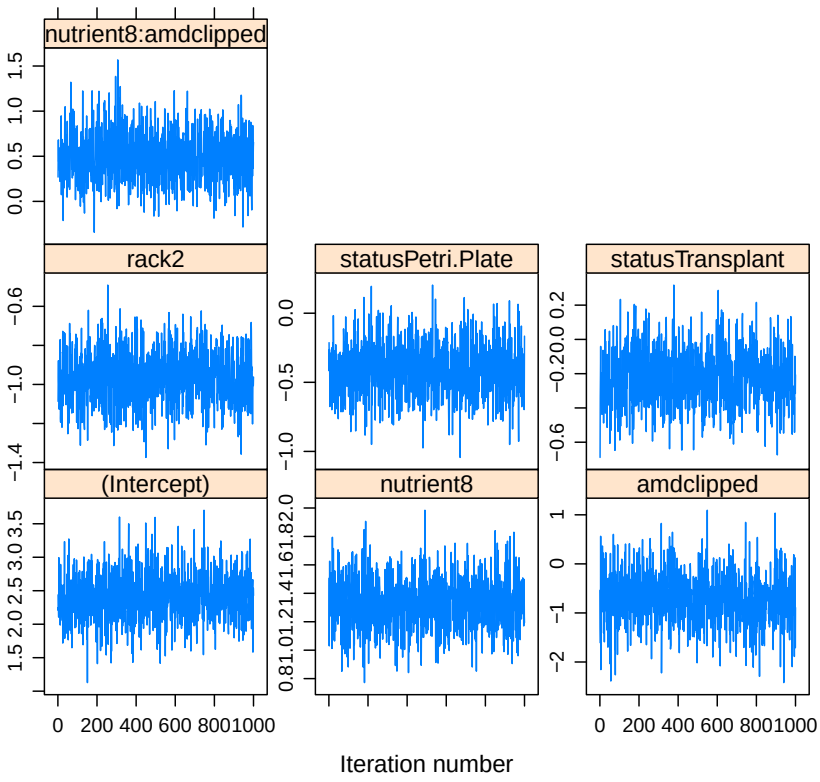

Figure 1: Trace plots for fixed effects in `MCMCglmm` fit of the full model.

```
> print(xyplot(as.mcmc(mcmc1$VCV),layout=c(3,3)))
```

Unfortunately, even in the simplest model the trace plots for the variances (not shown) don't look much better: the genotype-level variation still spends much of its time stuck at zero.[3]

[3]For completeness, we show here how to compare the fixed-effect parameters across the `MCMCglmm` fits and the original `glmer` fit, and how to compare variance parameters across `MCMCglmm` fits:

```
> coefplot2(list("MCMC 1"=mcmc1,
                 "MCMC 2"=mcmc2,
                 "MCMC 3"=mcmc3,
                 "glmer"=mp4),merge.names=FALSE,
            intercept=TRUE,
            legend=TRUE,
```

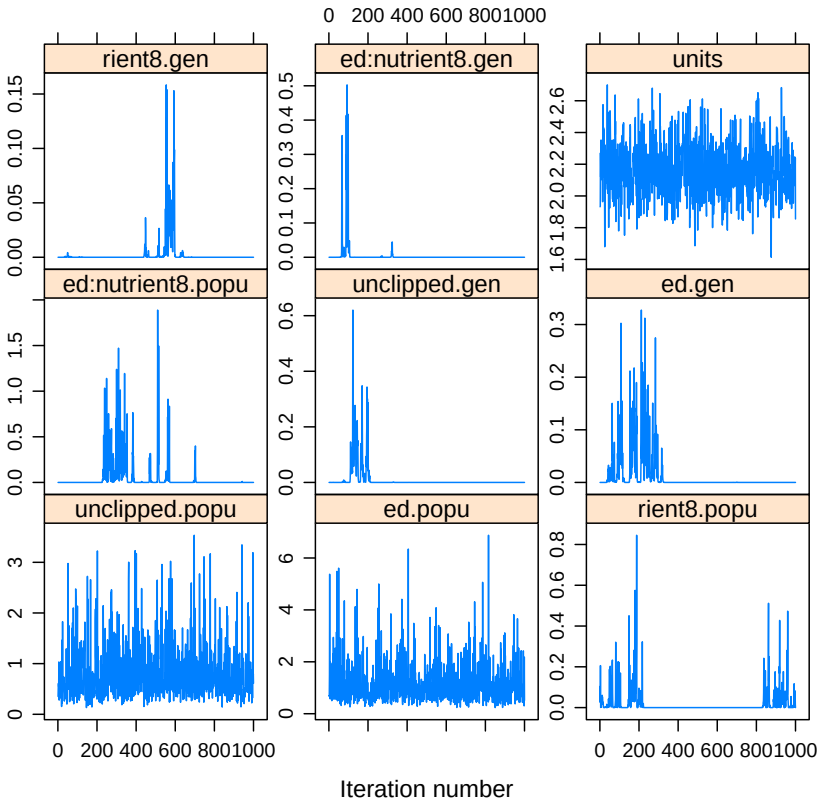

Figure 2: Trace plots for random effects in `MCMCglmm` fit of the full model.

To be confident in the results we would try (1) dropping the genotype-level variance term; (2) setting a weakly informative prior to push the es-

```
          legend.x="right")

> op <- par(mfrow=c(2,1))
> coefplot2(mcmc1,ptype="vcov",intercept=TRUE,
          varnames=c(outer(c("base","clip","nutr","inter"),
            c("pop","gen"),paste,sep="_"),"obs"))
> coefplot2(mcmc3,ptype="vcov",col="red",
          varnames=c("base_pop","base_gen","obs"),
          intercept=TRUE,
          top.axis=FALSE,main="",xlim=c(0,3))
> par(op)
```

timate of the genotype-level variance away from zero and improve mixing; (3) running the chain for much longer (e.g. setting `nitt=1e6`, `thin=1000`).

In general, we recommend `coefplot2` from the `coefplot2` package, along with `xyplot` and `densityplot` from the `coda` package, for graphical summaries, and `raftery.diag`, `geweke.diag`, `effectiveSize`, and `HPDinterval` from the `coda` package for numerical summaries of MCMC information.

3 glmmADMB

```
> detach("package:coefplot2")
> detach("package:lme4")
> library(glmmADMB)
> library(lme4) ## load lme4 AFTERWARDS
```

[4]

We run a batch file that fits NB1 and NB2 (i.e. linear- and quadratic-variance parameterizations of the negative binomial: see Part 1), with and without zero-inflation, using models such as

```
> gnb2 <- glmmadmb(total.fruits ~ nutrient*amd +
                                    rack + status,
                                    random=~(amd*nutrient|gen)+
                                    (amd*nutrient|popu),
                                    data=dat.tf, family="nbinom")
```

where we specify `family="nbinom1"` for NB1 fits and `zeroInflation=TRUE` for zero-inflated models. (In `glmmADMB` the random effects can be specified either in the same formula, as in the `lme4` package, or in a separate `random` argument, as in `nlme` or `MCMCglmm`.)

```
> load("Banta_glmmADMB_fits.RData")
```

The loaded R object `fits` is a list of fits: `gnb[12][Z]` specifies negative binomial type 1 (linear variance-mean relationship) or type 2 (quadratic variance-mean relationship), with or without zero-inflation: `gnb1B` is an additional model fit with nutrient effects at the genotype level and intercept effects at the population level, while `gnb1C` is our final model (`mp4` from Part 1: intercept variance at genotype and population levels).

[4] We would also have to `detach` the `doBy` package at this point, if it were already loaded; it requires `lme4`, so `lme4` can't be unloaded without unloading `doBy` first.

```
> AICtab(fits)

      dAIC  df
gnb1C   0.0 10
gnb1B   1.7 11
gnb1   11.6 16
gnb1Z  13.6 17
gnb2  134.6 16
gnb2Z 136.4 17
```

It looks like NB1 is much better than NB2 ($\approx$ 100 AIC units) ... but zero-inflation isn't doing anything.

Comparing the fixed-effect parameters (Figure 3:

```
> library(coefplot2)
> ff <- fits[c("gnb1","gnb2","gnb1B","gnb1C")]
> names(ff) <- c("NB1, full","NB2, full",
                 "NB1, nutxgen","NB1, int")
> coefplot2(ff,legend.x="right",legend=TRUE)
```

Figure 3 shows that the parameters are slightly different (NB2 estimates a stronger interaction effect), but not really qualitatively different — and we should go with the better-fitting model in any case.

It helps with, but doesn't entirely get rid of, the artifacts in the location-scale plot, though. Figure 4 compares the lognormal-Poisson fit from `glmer` (i.e., `mp4` from Part 1) with the NB1 fit from `glmmADMB`.

```
> op <- par(mfrow=c(1,2))
> locscaleplot(mp4)
> locscaleplot(fits$gnb1C)
> par(op)
```

4 via LMM approximation (`lme4::lmer`)

Would we have come to a significantly different answer via the traditional normal approximation? In general, we agree with those who argue that one should prefer modeling solutions that explicitly take the distribution of the data into account rather than transforming to achieve normality (O'Hara and Kotze, 2010; Warton and Hui, 2011), but since GLMMs can be significantly harder to fit than LMMs it is worth comparing the approaches.

We will again use the final model we settled on in part 1 (`mp4`, genotype- and population-level random intercepts):

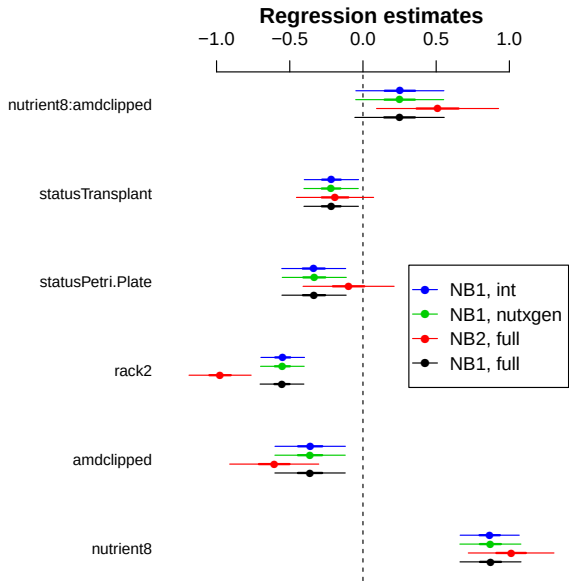

Figure 3: Comparison of fixed effect model estimates for models using linear type 1 (NB1) or quadratic type 2 (NB2) negative binomial distribution and either: all random effects (`full`); nutrient effects at genotype level, but intercept effects at population level (`nutxgen`); or the simplest model, with intercept effects estimated at both genotype and population levels (`int`).

```
> library(lme4) ## re-load lme4 package
> lm1 <- lmer(log(1+total.fruits)~nutrient*amd+rack+status+
            (1|popu)+(1|gen),data=dat.tf)
```

The location-scale plot (Figure 5) looks reasonable, although it again contains some artifacts based on the zero observations.

```
> locscaleplot(lm1,col=ifelse(dat.tf$total.fruits==0,"blue","black"))
```

The Q-Q plot (Figure 6) is more questionable.

```
> r <- residuals(lm1)
> qqnorm(r)
> qqline(r,col=2)
```

Neither the fixed nor the random effect variances are extremely different (Figure 7).

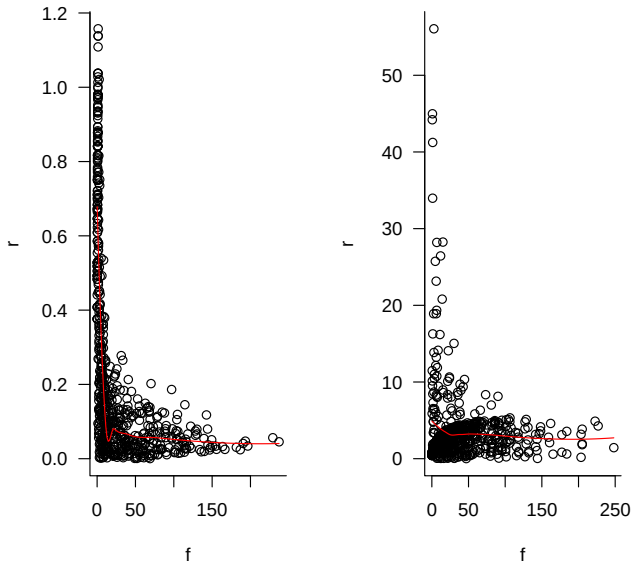

Figure 4: Location-scale plots of our final model from `glmer` (using a log-Poisson distribution) and from `glmmADMB` (using a type 1 negative binomial fit). Each plot has a superimposed loess fit.

```
> op <- par(mfrow=c(2,1))
> coefplot2(list(lm1,mp4))
> coefplot2(list(lmer=lm1,glmer=mp4),ptype="vcov",
            main="",intercept=TRUE,
            xlim=c(0,1.5),
            legend=TRUE)
> par(op)
```

We briefly considered testing to see whether the LMM or the GLMM fits the data better, but it is a little tricky (one has to add an additional factor to the log-likelihood to account for the $\log(1+x)$ transformation), and we decided not to try. As Jack Weiss comments in his course notes,

> The best recommendation I can make is not to [compare count-data models such as GLMMs to continuous-data models such as transformed LMMs]. The use of continuous distributions to model count data dates from a time when fitting discrete prob-

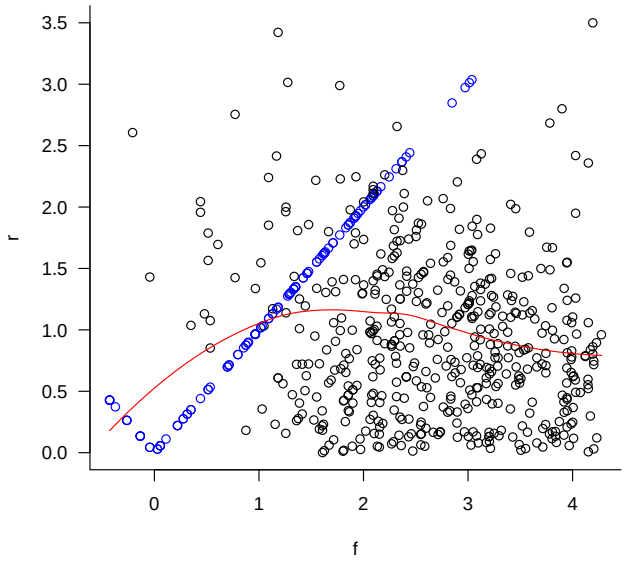

Figure 5: Location-scale plot of our final model via LMM approximation using `lmer`, with superimposed loess fit. Zeros in the data are marked in blue.

> ability distributions to data was difficult. Yet the use of the normal distribution as a probability model for count data or log-transformed count data is still widespread. Does it make sense to use AIC or log-likelihood to compare continuous probability models with discrete probability models?

He goes on to point out some fairly subtle difficulties involved in such comparisons.

5 Penalized quasi-likelihood (`MASS::glmmPQL`)

In the first version of this document, we used `family="quasipoisson"` in `glmer` to account for overdispersion. The `glmer` function no longer accepts `quasi` families, so we cannot directly compare our new approach with observation-level variance (Part 1), but we can use the `glmmPQL` function in the `MASS` package to test a quasi-likelihood approach.

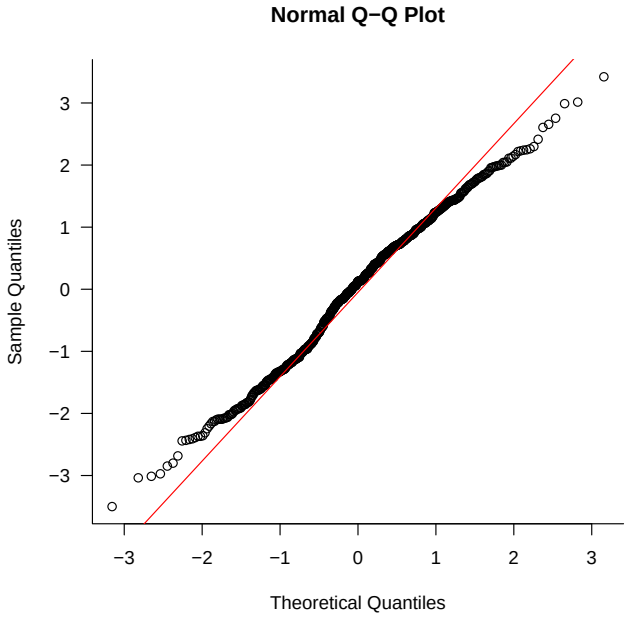

Figure 6: Q-Q plot of the residuals from a full model using a normal distribution on $\log(1+x)$-transformed data (`lmer`).

```
> library(MASS)

> mp1Q <- glmmPQL(total.fruits ~ nutrient*amd +
              rack + status,
                random=list(~amd*nutrient|popu,
                  ~amd*nutrient|gen),
                data=dat.tf, family="quasipoisson")
```

The variance-covariance structure is a bit unwieldy:

```
> VarCorr(mp1Q,rdig=2)
```

```
                    Variance                  StdDev       Corr
popu =              pdLogChol(amd * nutrient)
(Intercept)         2.168255e-01              4.656452e-01 (Int) amdcl ntrn8
amdclipped          3.124166e-07              5.589424e-04 0
nutrient8           1.084291e-07              3.292858e-04 0     0
amdclipped:nutrient8 2.774072e-07             5.266946e-04 0     0     0
```

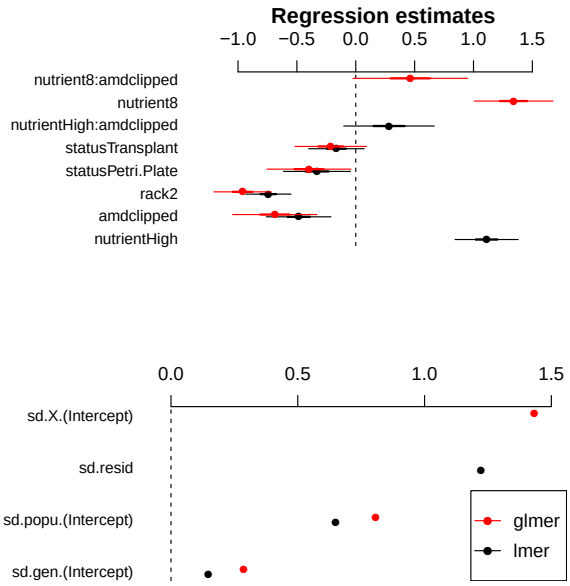

Figure 7: Regression estimates for fixed effects (top) and random effects estimates / predictions (bottom) for both LMM (black = `lmer`) and GLMM (red = `glmer`) fits. Also included are estimates of the observation-level standard deviation for each model (`SD.resid.` and `SD.X.(Intercept)`, respectively).

```
gen =                 pdLogChol(amd * nutrient)
(Intercept)           2.641394e-08              1.625237e-04 (Int) amdcl ntrn8
amdclipped            4.262859e-11              6.529058e-06  0.00
nutrient8             6.246685e-03              7.903597e-02 -0.17  0.01
amdclipped:nutrient8  1.224332e-06              1.106495e-03  0.00  0.00  0.00
Residual              3.032360e+01              5.506687e+00
```

The take home message is that most of the estimated variance components are very small: the three largest variance components are the residual variance ($\sigma^2 \approx 30$), population-intercept ($\sigma^2 \approx 0.22$), and variation in nutrient effect across genotypes ($\sigma^2 \approx 0.0062$). While formal significance testing of quasi-likelihood models is a bit hard because we do not have a log-likelihood or its derived quantities to work with, but we feel safe in asserting that this method does not allow us to estimate variation that the other methods failed

to discover. (Unfortunately, this also leads us to the conclusion that our previous results — from the first version of the supplementary online material, published in 2009 — based on quasi-likelihood methods with `glmer` were flawed.)

The fixed-effect coefficients are all similar.

```
> coefplot2(list(glmer=mp4,
                 MCMCglmmm=mcmc3,
                 lmer=lm1,
                 glmmPQL=mp1Q,
                 glmmADMB_NB1=fits$gnb1C),
           merge.names=FALSE,intercept=TRUE,
           legend.x="right",legend=TRUE)
```

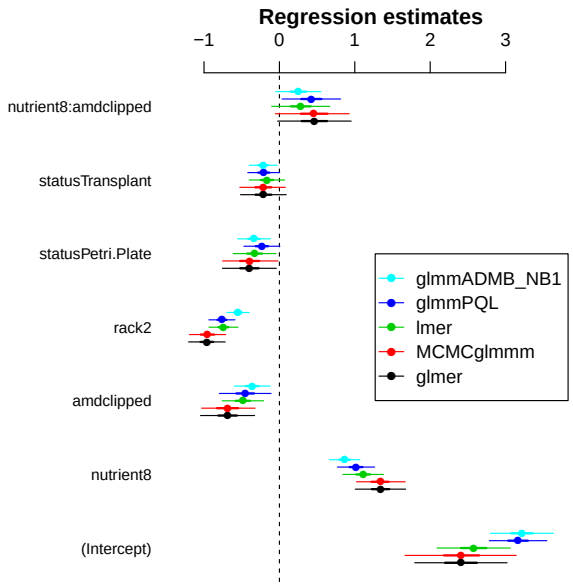

Figure 8: Summary of regression estimates of fixed effects for our final model from each of the major estimation methods discussed.

While the pattern is not perfectly consistent, Figure 8 shows that in general `glmmADMB` and `glmmPQL` results are most similar (both assume the variance increases linearly with the mean); the `lmer` fit is slightly different ($V \propto \mu^2$); and the `MCMCglmm` and `glmer` fits ($V \propto \mu(1 + C\mu)$) are similar

to each other. (Surprisingly, a negative binomial (NB2) fit from `glmmADMB` did not closely match the `glmer` and `MCMCglmm` fits, as we would have expected because it has the same mean-variance relationship.) However, these differences are really very small when considered in biological terms. In this example, if we had proceeded carefully we would have drawn the same biological conclusions no matter what model we started with:

- There is insufficient variation (alas!) to reliably quantify the variation in nutrient/clipping/interaction effects across genotypes and populations;
- Total fruit set varies both among genotypes within a population and among populations;
- nutrient has a positive effect, clipping has a negative effect, and adding nutrient appears to cancel out the effect of clipping.

References

Banta, J. A., M. H. H. Stevens, and M. Pigliucci. 2010. A comprehensive test of the 'limiting resources' framework applied to plant tolerance to apical meristem damage. Oikos **119**:359–369. URL `http://onlinelibrary.wiley.com/doi/10.1111/j.1600-0706.2009.17726.x/abstract`.

Bolker, B. M., M. E. Brooks, C. J. Clark, S. W. Geange, J. R. Poulsen, M. H. H. Stevens, and J.-S. S. White. 2009. Generalized linear mixed models: a practical guide for ecology and evolution. Trends in Ecology and Evolution **24**:127–135.

O'Hara, R. B. and D. J. Kotze. 2010. Do not log-transform count data. Methods in Ecology and Evolution **1**:118–122. URL `http://onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2010.00021.x/abstract`.

Warton, D. I. and F. K. C. Hui. 2011. The arcsine is asinine: the analysis of proportions in ecology. Ecology **92**:3–10. URL `http://www.esajournals.org/doi/full/10.1890/10-0340.1`.