Salamander Example Comparing GLMMs, Zero-Inflated GLMMs, and Hurdle Models

Mollie Brooks

2016-12-29

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 mountian top removal coal mining. The data was originally published in Price et al. (2016) and was aquired from Dryad (Price et al. (2015)).

Preliminaries

Load packages

```
library(glmmTMB)
library(ggplot2); theme_set(theme_bw())
library(knitr)
library(bbmle) #for AICtab
library(reshape)
library(plyr)
```

Load and organize data

```
data(Salamanders)
head(Salamanders)
```

```
##    site mined cover sample  DOP  Wtemp   DOY spp count
## 1 VF -1   yes -1.44      1 -0.6 -1.229 -1.50  GP     0
## 2 VF- 2   yes  0.30      1 -0.6  0.085 -1.50  GP     0
## 3 VF -3   yes  0.40      1 -1.2  1.014 -1.29  GP     0
## 4  R -1    no -0.45      1  0.0 -3.023 -2.71  GP     2
## 5  R -2    no  0.60      1  0.6 -0.144 -0.69  GP     2
## 6  R -3    no  1.34      1  0.6 -0.015 -0.69  GP     1
```

```
Salamanders = transform(Salamanders, present = as.numeric(count>0))
```

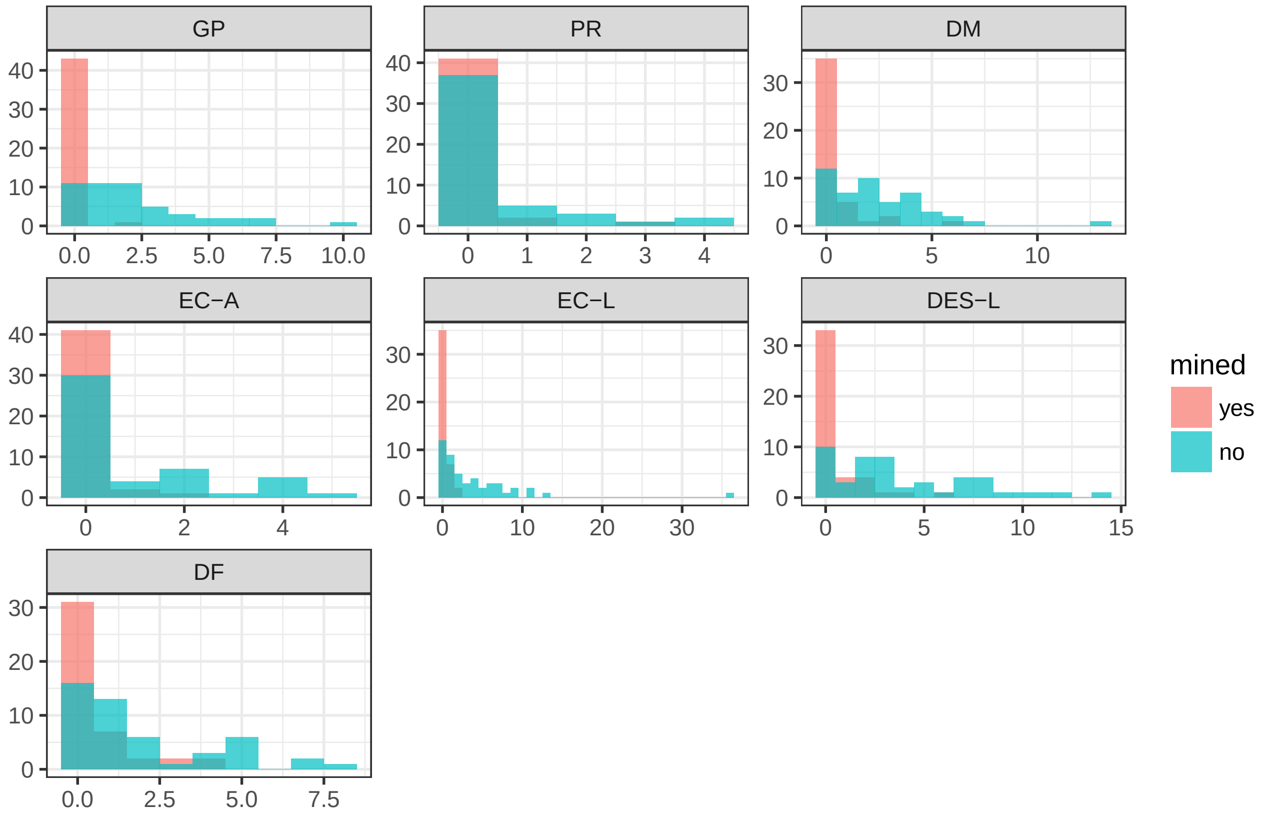

Figure 1 – Observed count data. Histograms of count data split into separate panels for each species or life stage. Each panel contains two overlaid histograms in which color represents whether the site was mined or not.

Model fitting with glmmTMB

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.

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")
```

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.

```
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")
```

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

```
## Warning in glmmTMB(count ~ spp + (1 | site), zi = ~spp, Salamanders, family
## = "nbinom2"): Model convergence problem. Hessian is not positive definite.
## This may indicate that a model is overparameterized.
```

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

```
## Warning in glmmTMB(count ~ spp + mined + (1 | site), zi = ~spp,
## Salamanders, : Model convergence problem. Hessian is not positive definite.
## This may indicate that a model is overparameterized.
```

```
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")
```

The warning messages tell us that `zinbm0` and `zinbm1` did not converge. However, the models with `mined` as a predictor of zero-inflation did converege.

Hurdle models

We can also fit hurdle models. This is done in two parts: first by modelling the zeros versus non-zeros with a binomial distribution and then modelling the non-zeros with a truncated distribution. In the salamander example, this means first modeling presence versus absence and then modeling the presence only data. It's important that the response `present` is numeric (`1` or `0`) instead of (`TRUE` or `FALSE`); that's why we wrote `present=as.numeric(count>0)` in the data organization code above.

```
zm0 = glmmTMB(present~spp, family="binomial", Salamanders)
zm1 = glmmTMB(present~spp + mined, family="binomial", Salamanders)
```

```
zm2 = glmmTMB(present~spp * mined, family="binomial", Salamanders)
AICtab(zm0, zm1, zm2)
```

```
##     dAIC  df
## zm2   0.0 14
## zm1   4.3 8
## zm0 167.2 7
```

For the second part of the hurdle model, we look at the presence data, `subset(Salamanders, present==1)`.

```
cpm0 = glmmTMB(count~spp + (1|site),
  family=list(family="truncated_poisson", link="log"),
  data=subset(Salamanders, present==1))
cpm1 = glmmTMB(count~spp + mined + (1|site),
  family=list(family="truncated_poisson", link="log"),
  data=subset(Salamanders, present==1))
cpm2 = glmmTMB(count~spp * mined + (1|site),
  family=list(family="truncated_poisson", link="log"),
  data=subset(Salamanders, present==1))
cnbm0 = glmmTMB(count~spp + (1|site),
  family=list(family="truncated_nbinom2", link="log"),
  data=subset(Salamanders, present==1))
cnbm1 = glmmTMB(count~spp + mined + (1|site),
  family=list(family="truncated_nbinom2", link="log"),
  data=subset(Salamanders, present==1))
cnbm2 = glmmTMB(count~spp * mined + (1|site),
  family=list(family="truncated_nbinom2", link="log"),
  data=subset(Salamanders, present==1))
AICtab(cpm0, cpm1, cpm2,
       cnbm0, cnbm1, cnbm2)
```

```
##       dAIC  df
## cnbm1   0.0 10
## cnbm2   2.1 16
## cnbm0  19.7 9
## cpm2  111.7 15
## cpm1  114.6 9
## cpm0  138.0 8
```

Model comparison using AIC

We can use `AICtab` to compare the GLMMs and zero-inflated GLMMs, but not hurdle models.

```
AICtab(pm0, pm1, pm2,
       nbm0, nbm1, nbm2,
       nbdm0, nbdm1, nbdm2,
       zipm0, zipm1, zipm2, zipm3,
       zinbm0, zinbm1, zinbm2, zinbm3)
```

```
##        dAIC  df
## nbm2     0.0 16
## nbdm2    1.5 17
## zinbm3   6.3 30
## zinbm2   6.9 18
## nbm1      9.0 10
## nbdm1     9.6 11
## nbm0     38.6 9
## nbdm0    39.0 10
## zipm3   114.7 29
## zipm2   122.1 17
## zipm1   139.6 16
## zipm0   169.1 15
## pm2     276.9 15
## pm1     299.5 9
## pm0     329.3 8
## zinbm0     NA 16
## zinbm1     NA 17
```

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 `zipm0$fit$objective` if it was needed.

To get the AIC value of hurdle models, we sum the AIC values of the two parts of the model. Then we can compare this AIC value to that of our best GLMM.

```
AIC(zm2)+AIC(cnbm1)
```

```
## [1] 1671
```

```
AIC(nbm2)
```

```
## [1] 1663
```

The negative binomial model has a slightly lower AIC value than the hurdle model.

Model summary

The summary of the negative binomial GLMM is similar to the familiar output from glmer, so we also present the summary from a more complicated model below to demonstrate output from zero-inflation and dispersion models.

```
summary(nbm2)
```

```
##  Family: nbinom2  ( log )
## Formula:          count ~ spp * mined + (1 | site)
## Data: Salamanders
##
##      AIC      BIC   logLik deviance df.resid
##     1663     1735     -816     1631      628
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  site   (Intercept) 0.284    0.533
## Number of obs: 644, groups:  site, 23
##
## Overdispersion parameter for nbinom2 family ():    1
##
## Conditional model:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.375      0.758   -4.45  8.4e-06 ***
## sppPR               0.931      0.877    1.06  0.28883
## sppDM               2.249      0.788    2.85  0.00431 **
## sppEC-A             0.714      0.905    0.79  0.43003
## sppEC-L             1.813      0.813    2.23  0.02574 *
## sppDES-L            2.511      0.779    3.22  0.00128 **
## sppDF               2.576      0.780    3.30  0.00096 ***
## minedno             4.162      0.793    5.25  1.5e-07 ***
## sppPR:minedno      -2.583      0.933   -2.77  0.00562 **
## sppDM:minedno      -2.150      0.826   -2.60  0.00925 **
## sppEC-A:minedno    -1.583      0.946   -1.67  0.09434 .
## sppEC-L:minedno    -1.338      0.849   -1.58  0.11510
## sppDES-L:minedno   -1.936      0.816   -2.37  0.01773 *
## sppDF:minedno      -2.743      0.822   -3.34  0.00084 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##  Family: nbinom2  ( log )
## Formula:          count ~ spp + mined + (1 | site)
## Zero inflation:         ~spp + mined
## Dispersion:             ~DOY
## Data: Salamanders
##
##      AIC      BIC   logLik deviance df.resid
##     1670     1755     -816     1632      625
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  site   (Intercept) 0.137    0.37
## Number of obs: 644, groups:  site, 23
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.6397     0.4096   -1.56  0.11835
## sppPR        -0.8280     0.5642   -1.47  0.14223
## sppDM         0.2202     0.2367    0.93  0.35222
## sppEC-A      -0.2308     0.3376   -0.68  0.49410
## sppEC-L       0.4278     0.2376    1.80  0.07177 .
## sppDES-L      0.6155     0.2253    2.73  0.00629 **
## sppDF        -0.0673     0.2445   -0.28  0.78300
## minedno       1.4180     0.3705    3.83  0.00013 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Zero-inflation model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.871      0.635    1.37    0.170
## sppPR          1.352      1.053    1.28    0.199
## sppDM         -0.865      0.787   -1.10    0.272
## sppEC-A        1.162      0.674    1.72    0.085 .
## sppEC-L       -0.633      0.744   -0.85    0.395
## sppDES-L      -0.890      0.752   -1.18    0.237
## sppDF         -2.396      2.012   -1.19    0.234
## minedno       -2.541      0.578   -4.39 0.000011 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Dispersion model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.433      0.245    1.77    0.077 .
## DOY           -0.252      0.195   -1.29    0.196
## ---
## 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 are only meaningful in comparison to other models fit by `glmmTMB`; this is because `glmmTMB` does not drop any constants from the likelihood while some other packages do. The second section describes the variability of the `Random effects`. In this model, we only had random effects on the conditional model, but random effects from the zero-inflation model could also appear here. The third section describes the coefficients of the `Conditional model` 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 unafected 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 unafected by mining). The last section describes the `Dispersion model`, which uses a log link to keep the dispersion parameter positive. This is in contrast to the summary of `nbm2` above, where there is no dispersion model and the overdispersion parameter is reported on the natural (rather than log) scale.

Plotting model results

As previously discussed in various places there are a whole bunch of decisions to make about marginalizing over or conditioning on the random effects. See discussion at this link.

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.

```r
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-1.98*temp$se.fit
newdata$predFE.max = temp$fit+1.98*temp$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")
```

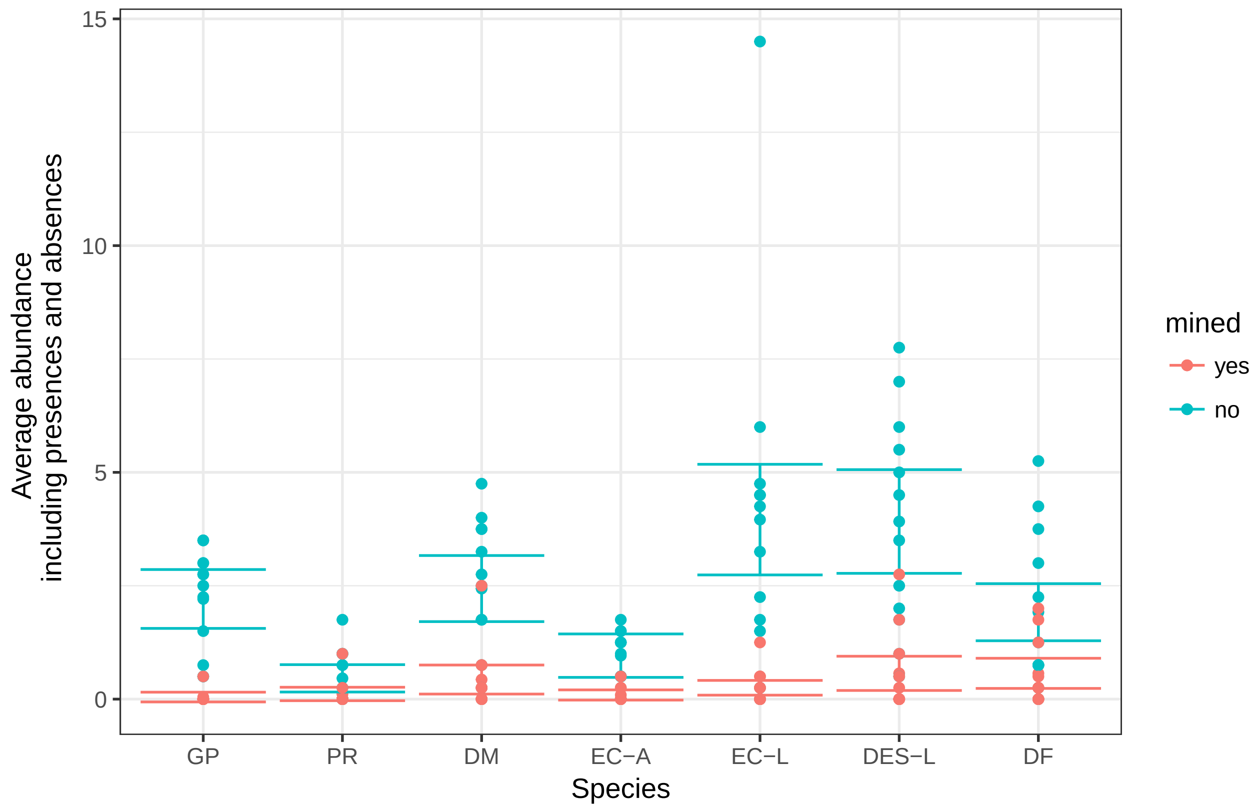

Figure 2 – Estimated abundance ignoring correlation. Points represtent site-specific average counts. Error bars represent the 95% Wald-type confidence intervals for the predicted average count.

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:

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

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

```r
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.

```r
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")
```

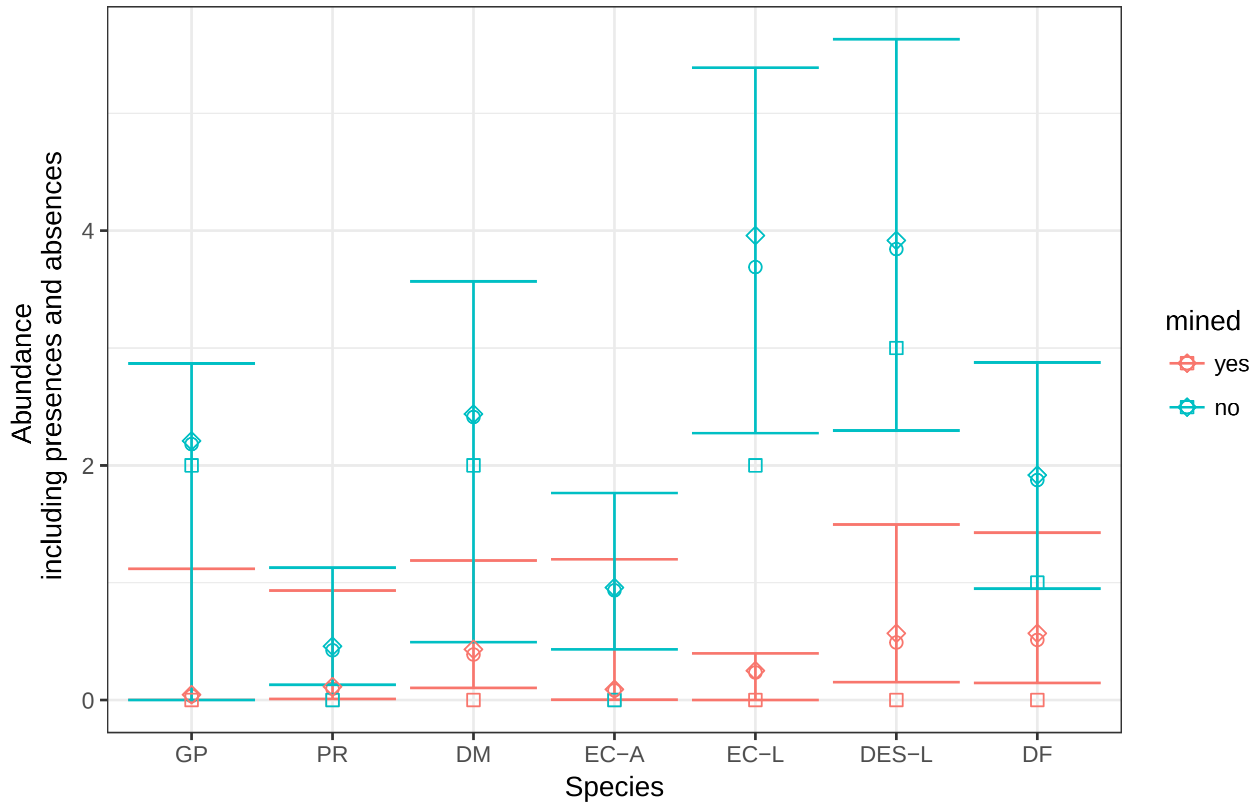

Figure 3 – Estimated abundance at mode. Circles represent predicted unconditional counts at the mode (i.e., site effect = 0) and error bars represent the 95% confidence intervals for that mode. Squares represnet the

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 look at the distribution of simulated values from the best fitted model. For this we use the function `simulate.glmmTMB`.

```
sims=simulate(zinbm3, 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)
```

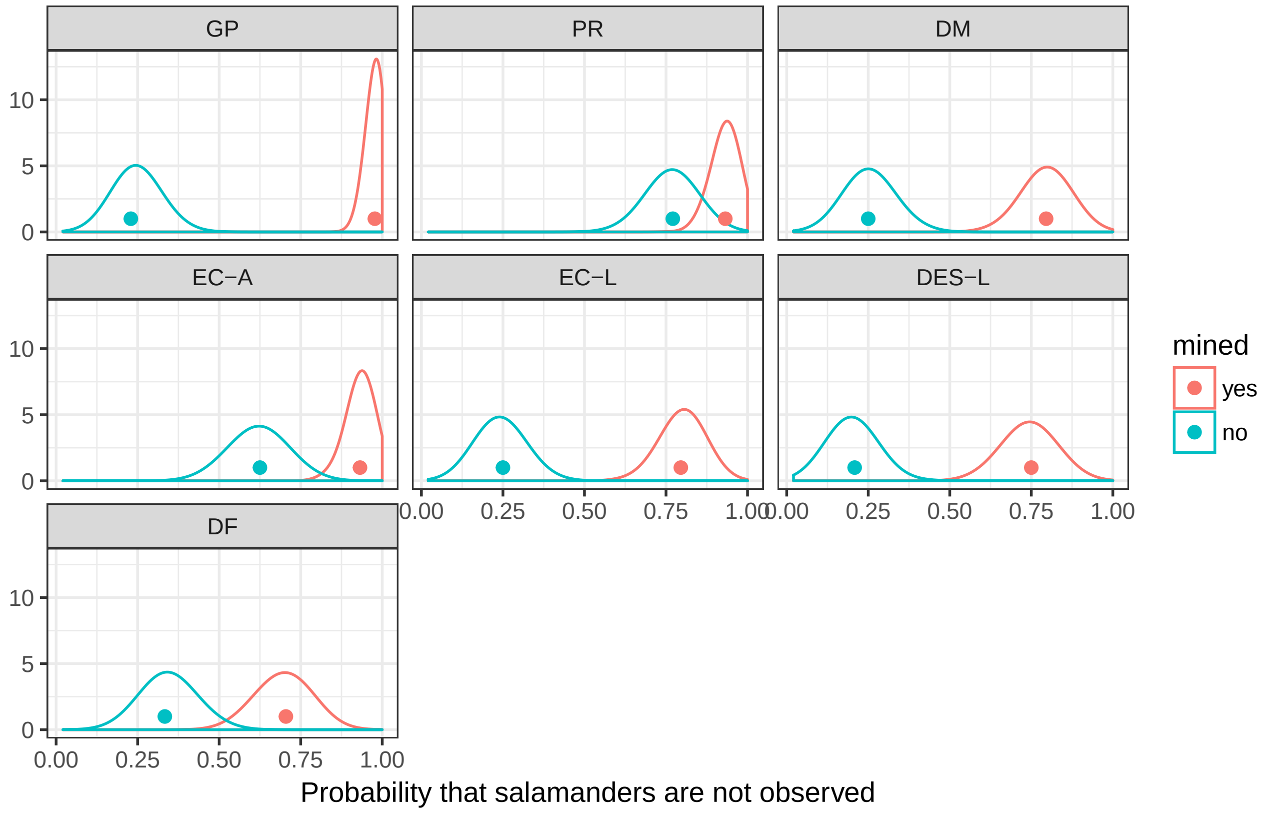

Figure 4 – Simulated zero counts. Each panel represents a different species or life stage of a species. Densities are values from 1000 data sets simulated from our best fit model. Points represent the observed data.

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

```r
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)
```

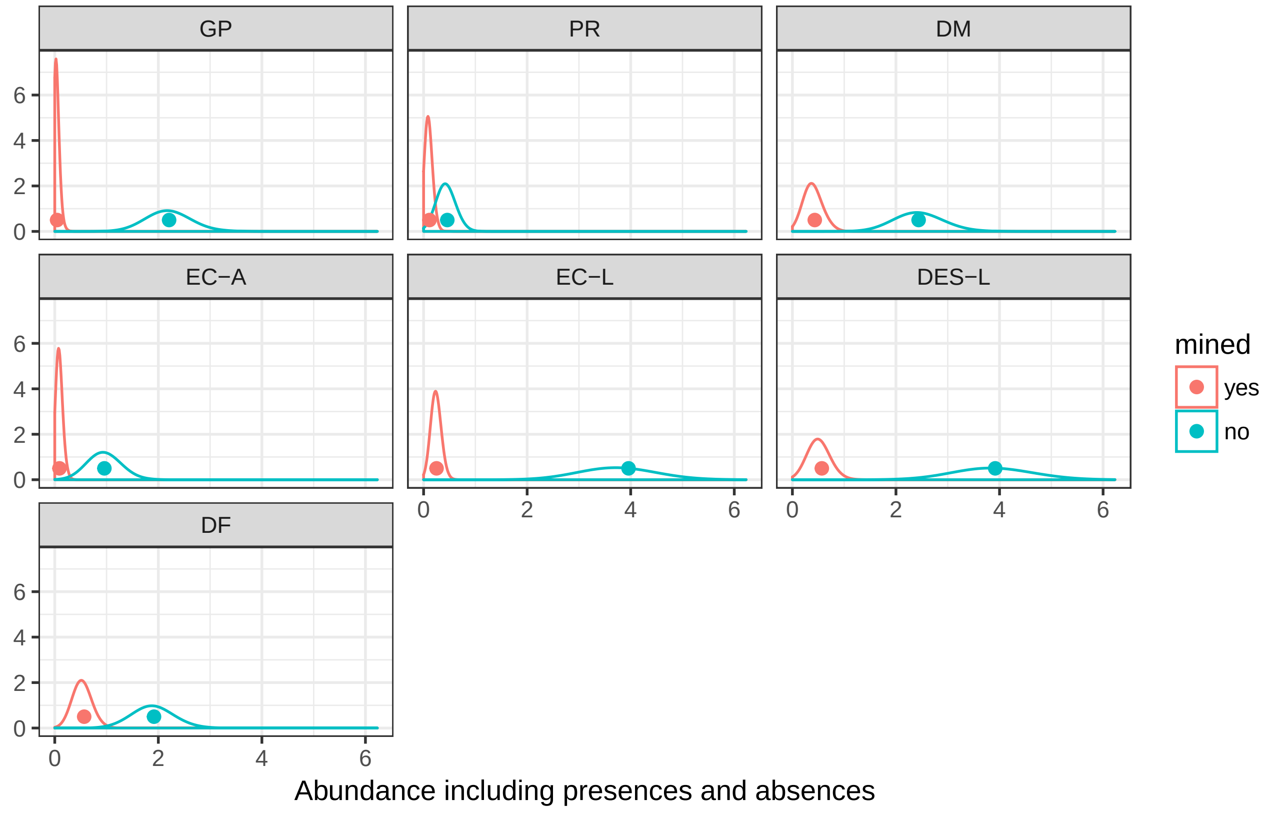

Figure 5 – Simulated unconditional abundances. Each panel represents a different species or life stage of a species. Densities are values from 1000 data sets simulated from our best fit model. Points represent the observed data.

References

Price, S J, B L Muncy, S J Bonner, A N Drayer, and C D Barton. 2015. "Data from: Effects of Mountaintop Removal Mining and Valley Filling on the Occupancy and Abundance of Stream Salamanders." *Journal of Applied Ecology*. Dryad Digital Repository. doi:doi:10.5061/dryad.5m8f6.

———. 2016. "Effects of Mountaintop Removal Mining and Valley Filling on the Occupancy and Abundance of Stream Salamanders." *Journal of Applied Ecology* 53 (2): 459–68. doi:10.1111/1365-2664.12585.