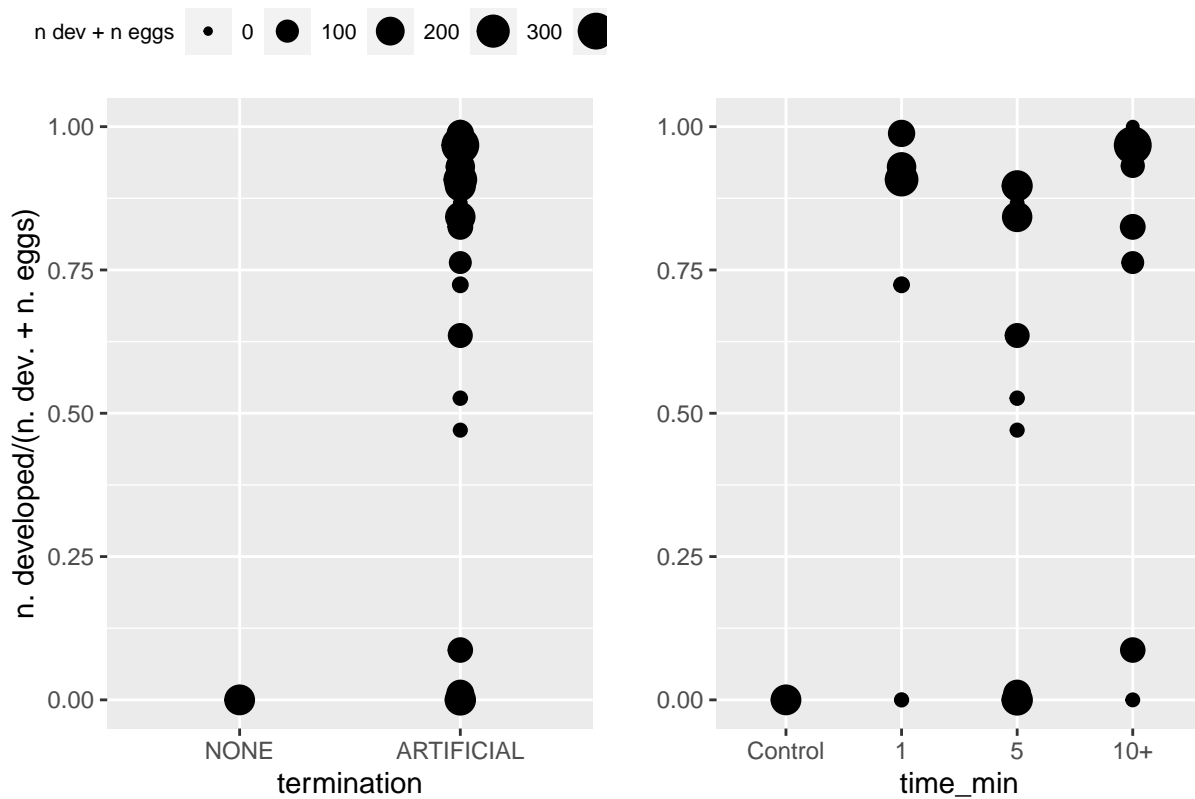


Fertilisation project - GLMM

SP

12/11/2020

The plot on the left is to illustrate the difference in the proportion of developed embryos between control (NONE) and all treatments combined (ARTIFICIAL). The right plot shows the same y variable but divided into the three different treatments. In both figures, the size of the points are proportional to the total number of eggs and embryos in each female (n dev + n eggs).



For now, let's forget about the different treatments (right plot) and just analyse the difference in proportions of developed embryos between control and all treatment combined (left plot). I have found three models that we can fit to the proportions of developed embryos.

1. A model without random effects that uses a binomial function and the sum of eggs and embryos (tot) in each female as weights:

```
fit_bin <- glm(formula = p_dev ~ termination, weights = tot, family = "binomial",  
               data = dd)
```

2. A mixed model that uses a binomial function with control and treatments (termination) as fixed effect and female ID as random effect (snail_ID):

```
fit_mixbin <- glmmTMB(formula = p_dev ~ termination + (1|snail_ID),
  data = dd,
  family = "binomial",
  control=glmmTMBControl(optimizer=optim, optArgs=list(method="BFGS")))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

3. A mixed model that uses a beta-binomial function for including overdispersion (quasi-binomial has been removed from GLMMs) with the same effects as in model 2:

```
fit_betabin <- glmmTMB(formula = p_dev ~ termination + (1|snail_ID),
  data = dd,
  family = "betabinomial",
  control=glmmTMBControl(optimizer=optim, optArgs=list(method="BFGS")))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
##           dAIC   df
## fit_betabin    0.0  4
## fit_mixbin    1.8  3
## fit_bin      1994.7 2
```

The beta-binomial model seems to fit the data slightly better than the other models and we can have a look at the summary of the model.

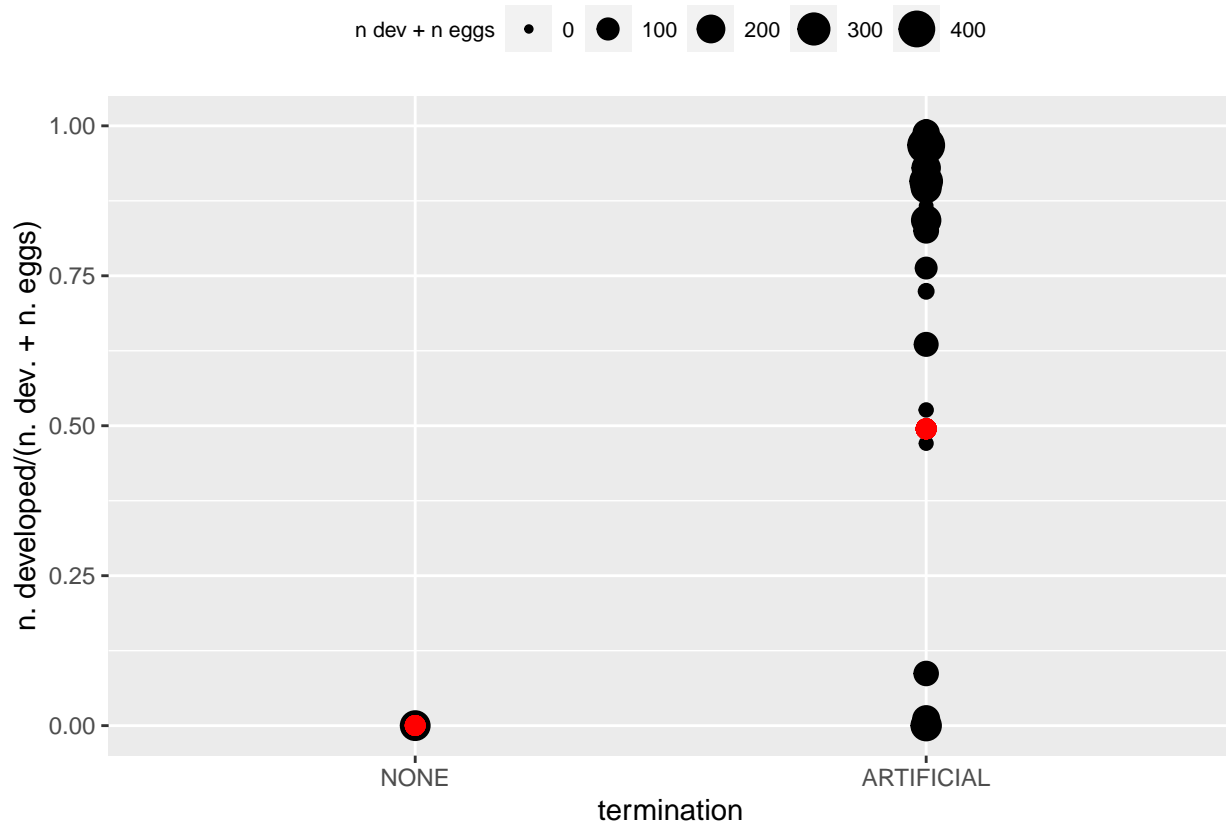
```
summary(fit_betabin)
```

```
## Family: betabinomial ( logit )
## Formula:          p_dev ~ termination + (1 | snail_ID)
## Data: dd
##
##      AIC      BIC   logLik deviance df.resid
##    38.9     44.6   -15.4     30.9       27
##
## Random effects:
##
## Conditional model:
## Groups   Name      Variance Std.Dev.
## snail_ID (Intercept) 1.797e-09 4.239e-05
## Number of obs: 31, groups:  snail_ID, 31
##
## Overdispersion parameter for betabinomial family (): 2.07e+04
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -8.296     25.857  -0.321   0.748
## terminationARTIFICIAL  8.276     25.860   0.320   0.749
```

The standard errors of the parameters are large and even if the estimates of control and all treatment combined (ARTIFICIAL) are essentially the opposite, there is no clear difference between the two in the proportions of developed embryos. I imagine that the SEs are so high because the y variable is very dispersed for the

treatments but I could not find another way to account for that than the use of a model with overdispersion. Perhaps I should somehow include the estimate of the overdispersion parameter which seems very high to me but I am not aware of other estimates of overdispersion.

The fitted values are either 0 or 0.5 (see red dots below) which are certainly not reliable.



Let's also have a look at the summary of the mixed binomial model, the second best model.

```
summary(fit_mixbin)
```

```
## Family: binomial ( logit )
## Formula:          p_dev ~ termination + (1 | snail_ID)
## Data: dd
##
##      AIC      BIC   logLik deviance df.resid
##    40.7    45.0   -17.3    34.7     28
##
## Random effects:
##
## Conditional model:
##  Groups   Name      Variance Std.Dev.
## snail_ID (Intercept) 1.377e-05 0.00371
## Number of obs: 31, groups:  snail_ID, 31
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -10.70     86.06  -0.124   0.901
## terminationARTIFICIAL    10.68     86.06   0.124   0.901
```

Here, the errors are even higher as expected given that this model did not fit the data as good as the beta-binomial.

For curiosity, I have tried to use the proportion of eggs as the y variable. I have run the same steps as above and as expected the best model is still the beta-binomial and nothing has really changed.

```
AICtab(fit_bin, fit_mixbin, fit_betabin)

##           dAIC   df
## fit_betabin    0.0  4
## fit_mixbin     1.8  3
## fit_bin      2002.7  2

summary(fit_betabin)

## Family: betabinomial ( logit )
## Formula:           p_egg ~ termination + (1 | snail_ID)
## Data: dd
##
##      AIC      BIC   logLik deviance df.resid
##    30.8    35.8   -11.4    22.8      22
##
## Random effects:
##
## Conditional model:
##   Groups   Name      Variance Std.Dev.
## snail_ID (Intercept) 0.0001032 0.01016
## Number of obs: 26, groups:  snail_ID, 26
##
## Overdispersion parameter for betabinomial family (): 1.18e+06
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      9.704     52.274   0.186   0.853
## terminationARTIFICIAL -10.188     52.276  -0.195   0.845
```

I would not go for either of these models unless there is something I am doing wrong. Another method that we can try is to use counts and fit Poisson models but I cannot see why they would give a different result. Do you have other ideas? Shall we stick to this method?