Lathyrus ms3: Selective agents

Contents

```
# Load previously saved large objects
load("output/b_par_1.RData")
load("output/b_par_2.RData")
```

Data preparation

Load data, keep variables needed and merge

```
data_selag<-read.table("C:/Users/user/Dropbox/SU/Projects/lathyrus/lathyrus_ms1/data/clean/alldata_weat
mean_weather<-read.table("C:/Users/user/Dropbox/SU/Projects/lathyrus/lathyrus_ms1/data/clean/mean_weath
```

```
data_selag<-data_selag[c(1:7,9:10,12,14:15,17:18,21,22)]
data_selag$n_fl<-data_selag$cum_n_fl
data_selag$cum_n_fl<-NULL
mean_weather<-mean_weather[c(1,115:118)]
data_selag<-merge(data_selag,mean_weather,by="year")%>%
  arrange(id)%>%
  anti_join(subset(data_selag,is.na(FFD)&is.na(grazing)&
                     is.na(shoot_vol)&is.na(n_fr)&is.na(n_ovules)&
                     is.na(n_seeds)&is.na(n_intact_seeds)&
                     is.na(n_fl)))%>% # Remove rows with all these NAs
  filter(year!=1995) # Remove data from 1995 because of problems with predation
names(data_selag)
```

```
##  [1] "year"           "FFD"            "id"             "ruta"
##  [5] "genet"          "data"           "vernal"         "grazing"
##  [9] "shoot_vol"      "n_fr"           "n_ovules"       "FFD_corr"
## [13] "period"         "n_seeds"        "n_intact_seeds" "n_fl"
## [17] "mean_3"         "mean_4"         "mean_5"         "mean_6"
```

```
head(data_selag)
```

```
##   year      FFD     id ruta genet data              vernal    grazing shoot_vol
## 1 2006 60.00286  new_1 <NA>    NA    1 2006-03-20 18:25:00 1.00000000   1830.44
## 2 2006 58.27431 new_10 <NA>    NA    1 2006-03-20 18:25:00 0.37500000  21975.75
## 3 2007 43.03194 new_10 <NA>    NA    1 2007-03-21 00:14:00 0.00000000  19662.95
## 4 2008 44.78889 new_10 <NA>    NA    1 2008-03-20 06:04:00 0.07142857  28529.38
## 5 2009 49.54653 new_10 <NA>    NA    1 2009-03-20 11:53:00 0.92857143   1000.85
## 6 2010 57.30417 new_10 <NA>    NA    1 2010-03-20 17:42:00 0.00000000   4885.60
```

```
##   n_fr n_ovules   FFD_corr period   n_seeds n_intact_seeds n_fl     mean_3
## 1    0   0.0000       <NA>    new   0.00000     0.00000000   NA -3.6596774
## 2   25 324.0000 2006-05-18    new 111.00000    43.48453608  180 -3.6596774
## 3    5  62.0000 2007-05-03    new  22.00000     7.00000000  290  4.1000000
## 4   18 234.5143 2008-05-04    new  89.48571    89.48571429  156  1.5870968
## 5    0   0.0000 2009-05-09    new   0.00000     0.00000000  180  0.8661290
## 6    3  32.0000 2010-05-17    new   8.00000     0.04532821   28 -0.4612903
##     mean_4   mean_5   mean_6
## 1 4.611667 10.46613 15.96333
## 2 4.611667 10.46613 15.96333
## 3 7.203333 10.94677 15.94333
## 4 6.656667 11.46935 15.54500
## 5 7.186667 11.08387 13.59667
## 6 5.285000 10.70000 15.08667
```

List of variables in data set:

- year
- FFD: first flowering date (as number of days from vernal equinox)
- id: individual identifier (including "old" for individuals in period 1987-1996 and "new" for individuals in period 2006-2017)
- ruta, genet: identifiers for plots and ids in old data
- data: 1 if data available, 0 if not
- vernal: date of vernal equinox in each year
- grazing: proportion of grazing by deer
- shoot_vol: shoot volume
- n_fr: number of fruits
- n_ovules: number of ovules
- FFD_corr: first flowering date (as a date)
- period: "old" for 1987-1996 and "new" for 2006-2017
- n_seeds: number of seeds
- n_intact_seeds: number of intact (non-predated) seeds
- n_fl: number of flowers
- mean_3/4/5/6: average of daily mean temperatures for March/April/May/June

Interactions that we will focus on:

- Seed predation: proportion of seeds escaping predation
- Grazing (by deer) before flowering: proportion of grazing

Calculate proportion of predated seeds

```
data_selag<-data_selag%>%
  mutate(prop_pred_seeds=ifelse(n_seeds==0,NA,(n_seeds-n_intact_seeds)/n_seeds),
         n_pred_seeds=n_seeds-n_intact_seeds)
```

Using only mean temperatures. Using grazing as a proportion, and for 2008-2015 use values of proportion of aboveground volume. - 1987-1996: grazing = proportion of flowers removed - 2006: grazing = proportion of grazed shoots - 2007-2015: grazing = proportion of aboveground volume removed - 2016-2017: grazing = proportion of flowers removed

Calculate successes/failures for grazing (and weights)

```
data_selag$grazing_success<-round(with(data_selag,
                                       ifelse(is.na(grazing_corr),NA,
                                              ifelse(year<1997|year>2015,
                                                     grazing_corr*n_fl,
                                                     ifelse(year>2006&year<2016,
                                                            grazing_corr*shoot_vol,
                                                            999)))))
data_selag$grazing_failure<-round(with(data_selag,
                                       ifelse(is.na(grazing_corr),NA,
                                              ifelse(year<1997|year>2015,
                                                     n_fl-grazing_success,
                                                     ifelse(year>2006&year<2016,
                                                            shoot_vol-grazing_success,
                                                                 999)))))
data_selag$grazing_weights<-round(with(data_selag,
                                       ifelse(is.na(grazing_corr),NA,
                                              ifelse(year<1997|year>2015,
                                                     n_fl,
                                                     ifelse(year>2006&year<2016,
                                                            shoot_vol,
                                                            999)))))

grazing_success_2006<-read.table("data/grazing_success_2006.csv",
                                 header=T,sep=",",dec=".")
grazing_success_2006<-grazing_success_2006%>%
  mutate(weights=gr_success+gr_failure)
data_selag<-data_selag%>%
  left_join(grazing_success_2006)
data_selag$grazing_success<-with(data_selag,
                                 ifelse(year==2006,gr_success,grazing_success))
data_selag$grazing_failure<-with(data_selag,
                                 ifelse(year==2006,gr_failure,grazing_failure))
data_selag$grazing_weights<-with(data_selag,
                                 ifelse(year==2006,weights,grazing_weights))
data_selag$gr_success<-NULL
data_selag$gr_failure<-NULL
data_selag$weights<-NULL
head(data_selag)
```

```
##   year      FFD     id ruta genet data              vernal    grazing shoot_vol
## 1 2006 60.00286  new_1 <NA>    NA    1 2006-03-20 18:25:00 1.00000000   1830.44
## 2 2006 58.27431 new_10 <NA>    NA    1 2006-03-20 18:25:00 0.37500000  21975.75
## 3 2007 43.03194 new_10 <NA>    NA    1 2007-03-21 00:14:00 0.00000000  19662.95
## 4 2008 44.78889 new_10 <NA>    NA    1 2008-03-20 06:04:00 0.07142857  28529.38
## 5 2009 49.54653 new_10 <NA>    NA    1 2009-03-20 11:53:00 0.92857143   1000.85
## 6 2010 57.30417 new_10 <NA>    NA    1 2010-03-20 17:42:00 0.00000000   4885.60
##   n_fr n_ovules   FFD_corr period   n_seeds n_intact_seeds n_fl     mean_3
## 1    0   0.0000       <NA>    new   0.00000     0.00000000   NA -3.6596774
## 2   25 324.0000 2006-05-18    new 111.00000    43.48453608  180 -3.6596774
## 3    5  62.0000 2007-05-03    new  22.00000     7.00000000  290  4.1000000
## 4   18 234.5143 2008-05-04    new  89.48571    89.48571429  156  1.5870968
## 5    0   0.0000 2009-05-09    new   0.00000     0.00000000  180  0.8661290
## 6    3  32.0000 2010-05-17    new   8.00000     0.04532821   28 -0.4612903
```

```
##      mean_4   mean_5   mean_6 prop_pred_seeds n_pred_seeds grazing_new
## 1 4.611667 10.46613 15.96333              NA     0.000000          NA
## 2 4.611667 10.46613 15.96333       0.6082474    67.515464          NA
## 3 7.203333 10.94677 15.94333       0.6818182    15.000000         0.0
## 4 6.656667 11.46935 15.54500       0.0000000     0.000000         0.1
## 5 7.186667 11.08387 13.59667              NA     0.000000         0.9
## 6 5.285000 10.70000 15.08667       0.9943340     7.954672         0.0
##   grazing_corr grazing_success grazing_failure grazing_weights
## 1        1.000               1               0               1
## 2        0.375               3               5               8
## 3        0.000               0           19663           19663
## 4        0.100            2853           25676           28529
## 5        0.900             901             100            1001
## 6        0.000               0            4886            4886
```

H1 and H2: path model

H1: Flowering phenology of individuals influences the intensities of interactions with seed predators and grazers.

H2: These biotic interactions have important effects on fitness, and alter selection on flowering time within years.

Models include the effects of FFD and n_fl standardized within years (FFD_s_y and n_fl_s_y), grazing and seed predation on fitness relativized within years (fitness_rel_y), as well as the effects of FFD_s_y and n_fl_s_y on grazing and seed predation.

Calculate FFD_s_y, n_fl_s_y and fitness_rel_y:

```
data_selag<-data_selag%>%
  group_by(year) %>%
  mutate(FFD_mean=mean(FFD,na.rm=T),
         FFD_sd=sd(FFD,na.rm=T),
         FFD_s_y=(FFD-FFD_mean)/FFD_sd,
         n_fl_mean=mean(n_fl,na.rm=T),
         n_fl_sd=sd(n_fl,na.rm=T),
         n_fl_s_y=(n_fl-n_fl_mean)/n_fl_sd,
         fitness_mean=mean(n_intact_seeds,na.rm=T),
         fitness_rel_y=n_intact_seeds/fitness_mean)
```

```
data_selag$seeds_01<-with(data_selag,ifelse(is.na(n_seeds),NA,
                                            ifelse(n_seeds>0,1,0)))
```

```
path1_mod1<-glmmTMB(seeds_01~FFD_s_y+n_fl_s_y+grazing_corr+(1|id)+(1|year),
                data=data_selag,family="binomial")
path1_mod2<-glmmTMB(fitness_rel_y~FFD_s_y+n_fl_s_y+grazing_corr+prop_pred_seeds+
                   (1|id)+(1|year),data=data_selag)
path1_mod3<-glmmTMB(grazing_corr~FFD_s_y+n_fl_s_y+(1|id)+(1|year),
                  data=data_selag,family="binomial",weights=grazing_weights)
path1_mod4<-glmmTMB(prop_pred_seeds~FFD_s_y+n_fl_s_y+(1|id)+(1|year),
                  data=data_selag,family="binomial",
                  weights=round(n_seeds))
```

```r
nobs(path1_mod1)
```

```
## [1] 2376
```

```r
nobs(path1_mod2)
```

```
## [1] 1281
```

```r
nobs(path1_mod3)
```

```
## [1] 2354
```

```r
nobs(path1_mod4)
```

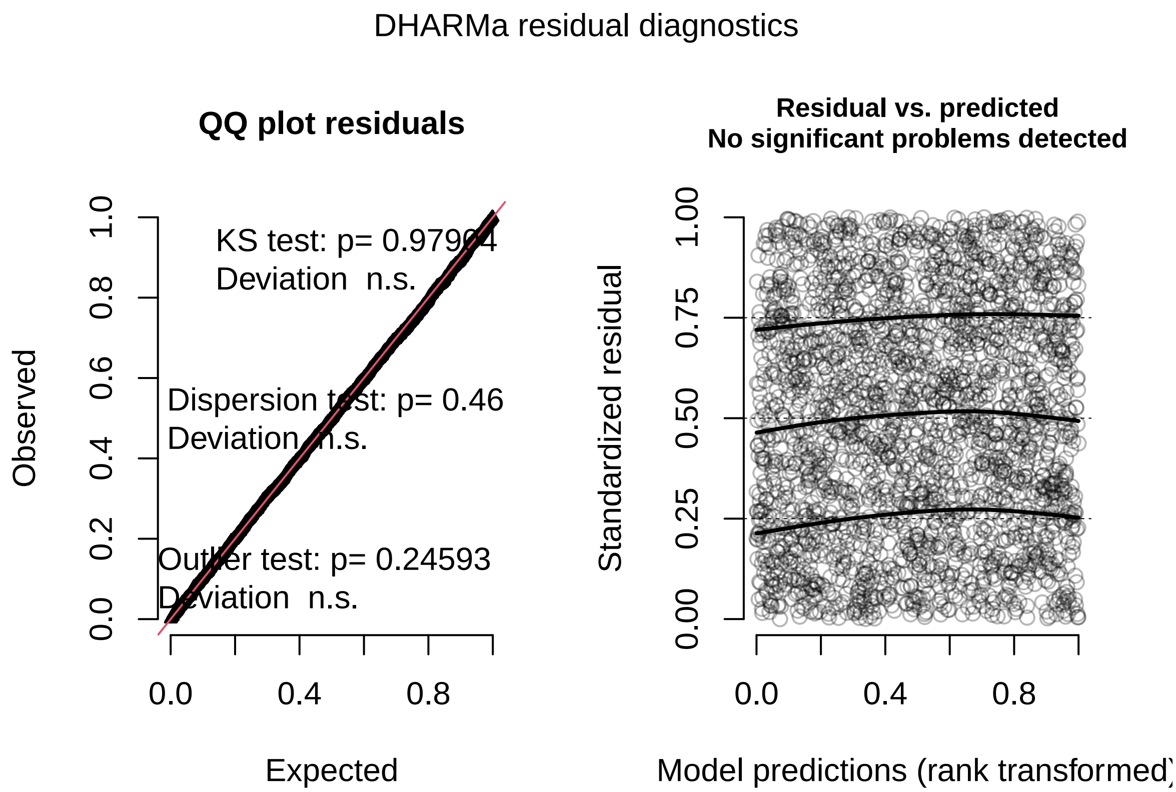
```
## [1] 1281
```

Model diagnostics

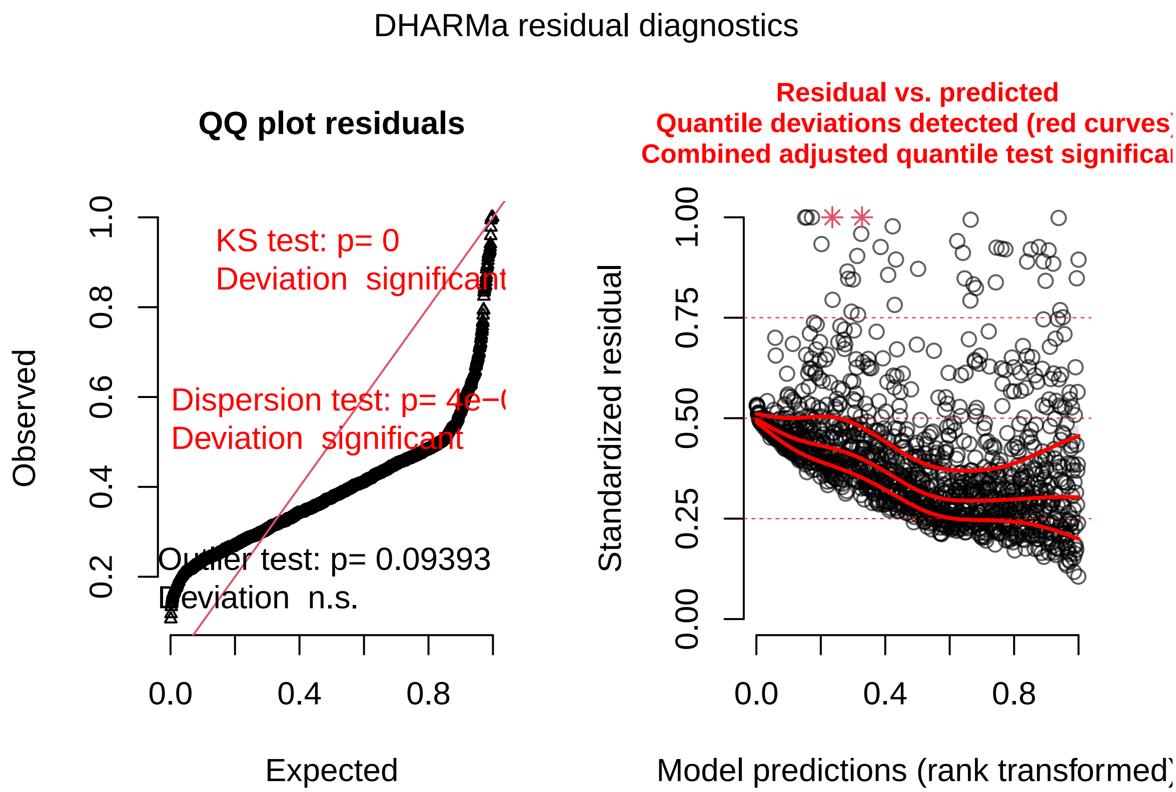
```r
sim_path1_mod1 <- simulateResiduals(fittedModel = path1_mod1, n = 5000)
sim_path1_mod2 <- simulateResiduals(fittedModel = path1_mod2, n = 5000)
sim_path1_mod3 <- simulateResiduals(fittedModel = path1_mod3, n = 5000)
sim_path1_mod4 <- simulateResiduals(fittedModel = path1_mod4, n = 5000)
```

qq-plot and plot of residuals vs. predicted:

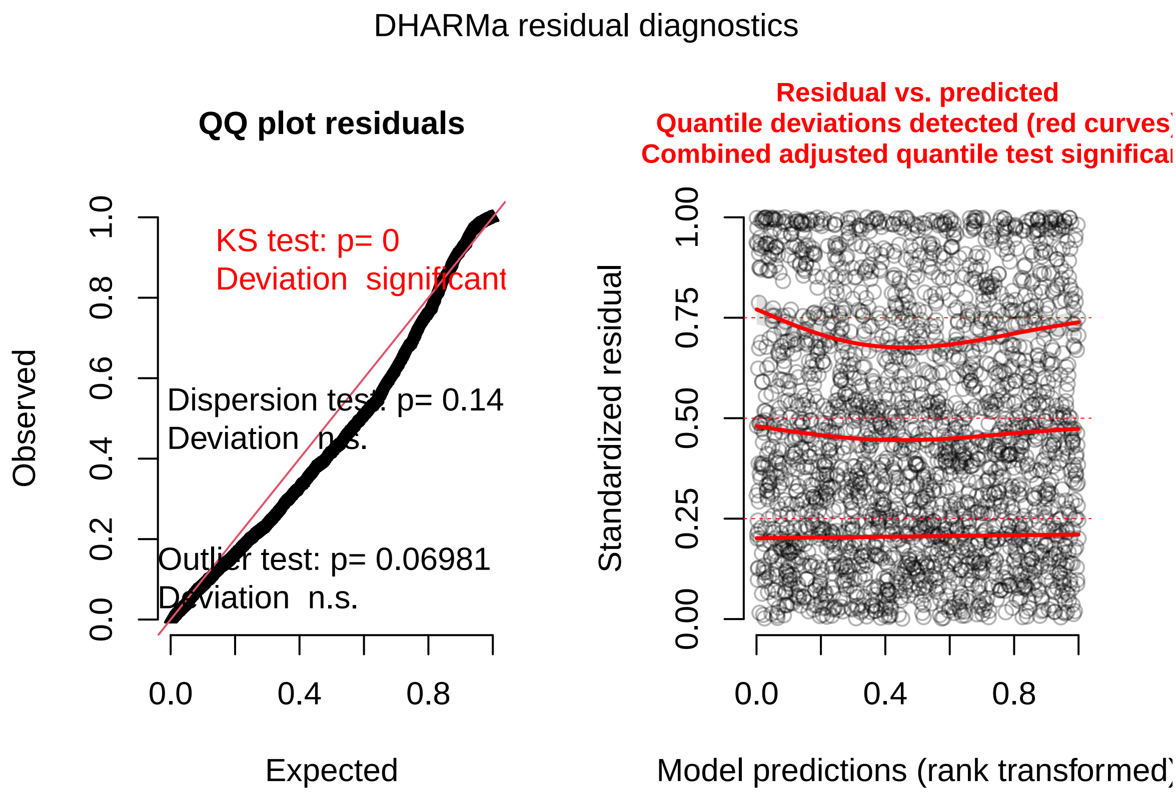
```r
plot(sim_path1_mod1,quantreg=T) # model OK
```



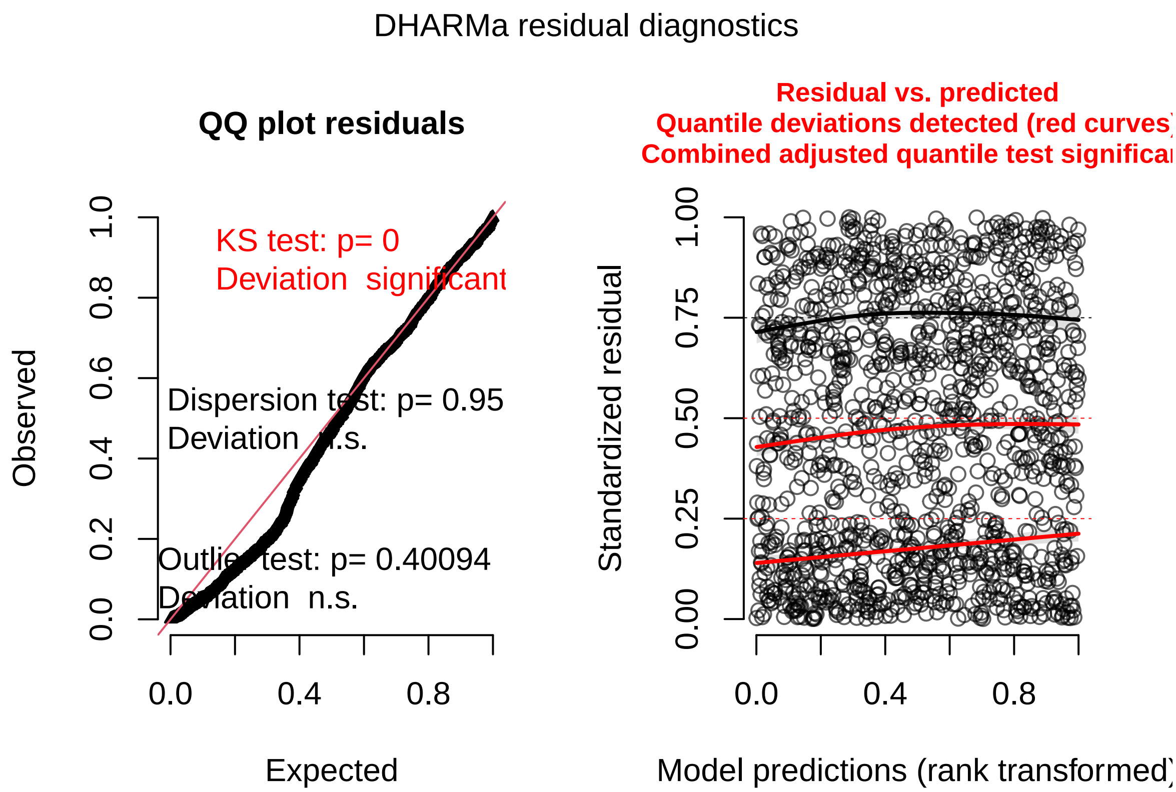

```
plot(sim_path1_mod2,quantreg=T) # not OK - calculate BCA intervals
```



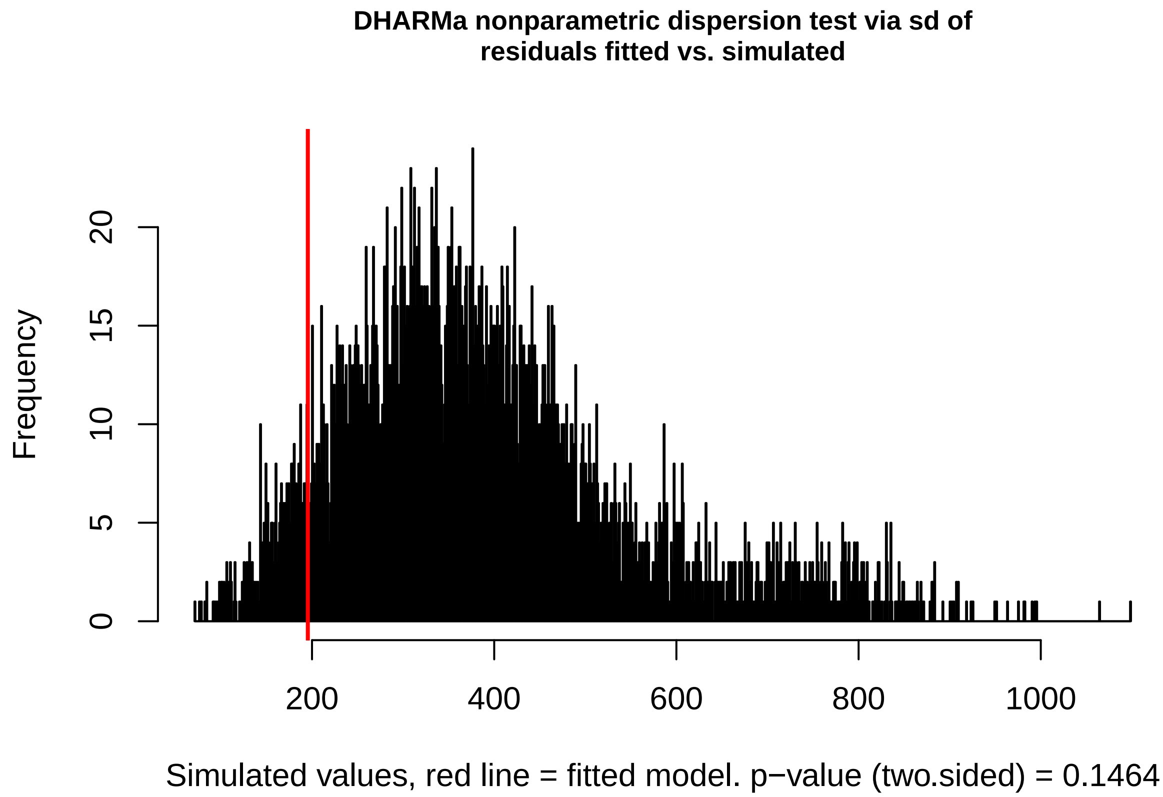

```
plot(sim_path1_mod3,quantreg=T) # not OK
```




```
plot(sim_path1_mod4,quantreg=T) # not OK
```

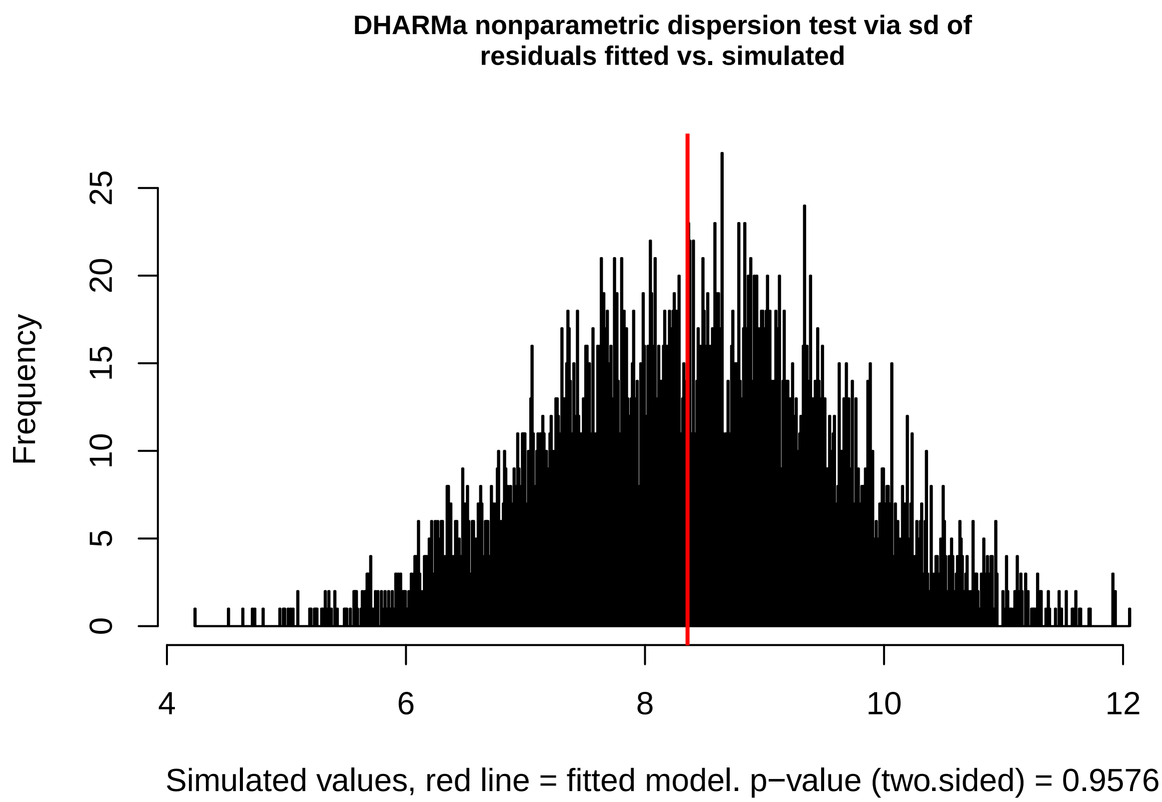



```r
testDispersion(sim_path1_mod3) # OK
```



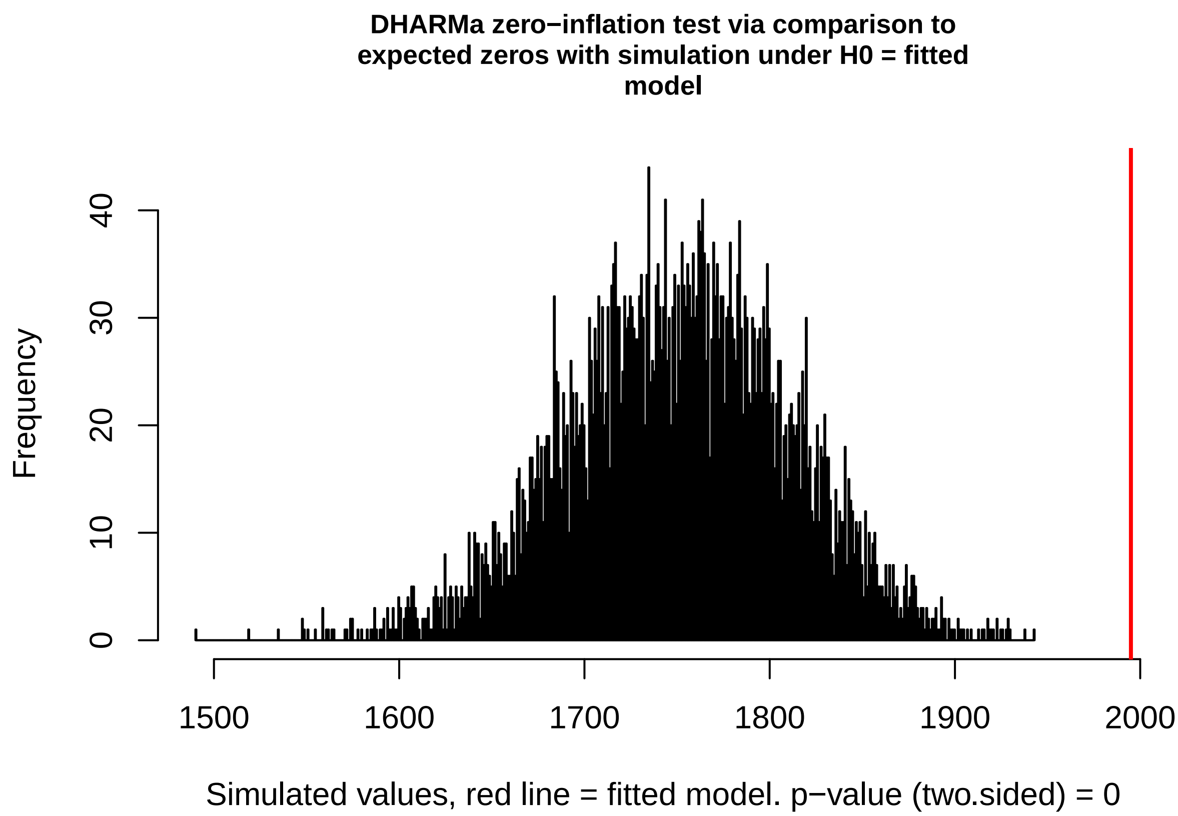

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.50412, p-value = 0.1464
## alternative hypothesis: two.sided
```

```
testDispersion(sim_path1_mod4) # OK
```



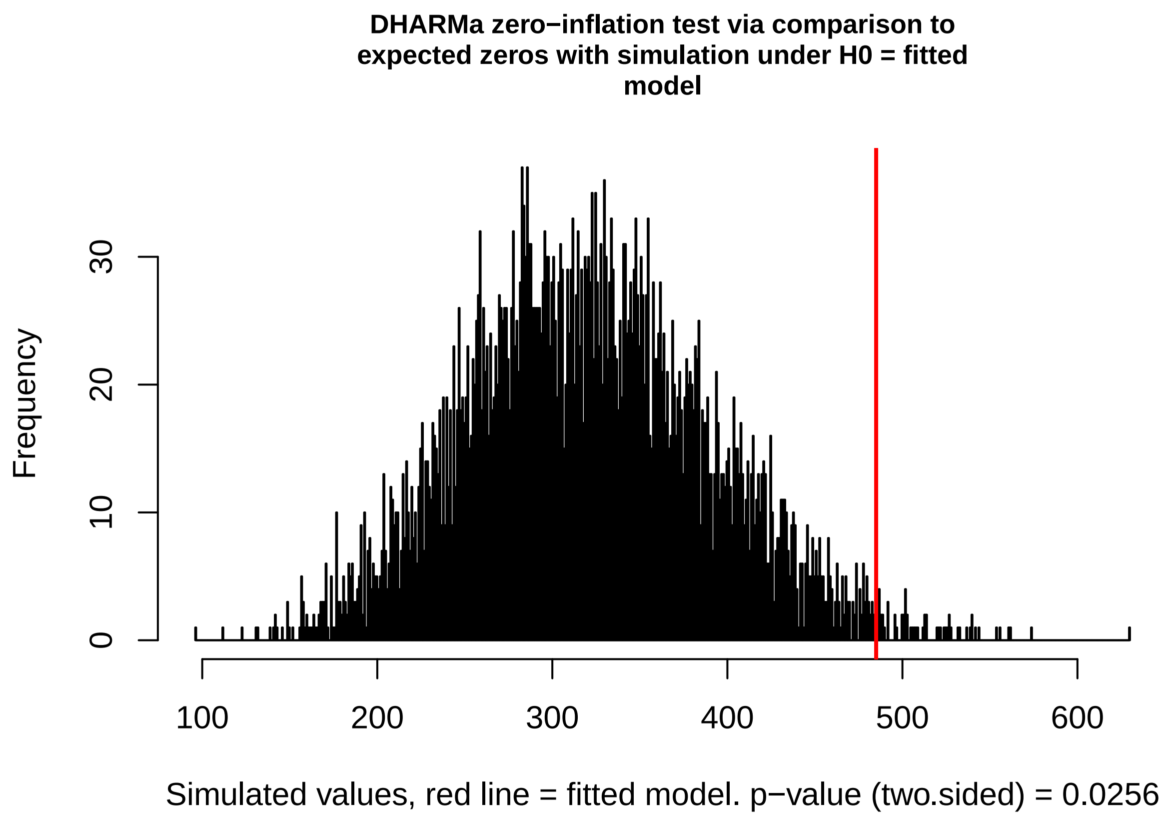

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.99474, p-value = 0.9576
## alternative hypothesis: two.sided
```

```
testZeroInflation(sim_path1_mod3) # significant
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.1394, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
testZeroInflation(sim_path1_mod4) # significant
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.5119, p-value = 0.0256
## alternative hypothesis: two.sided
```

Refit models 3 and 4 with betabinomial distribution.

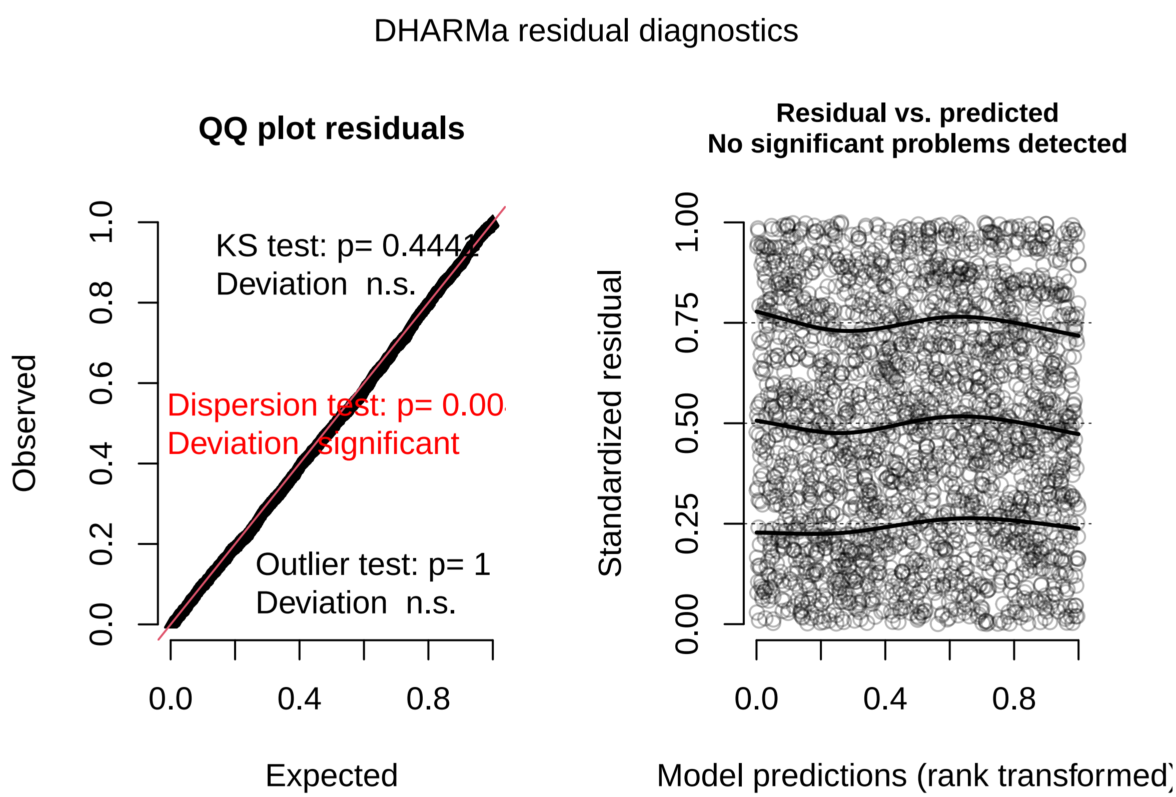
```
path1_mod3_bb<-glmmTMB(grazing_corr~FFD_s_y+n_fl_s_y+(1|id)+(1|year),
                            data=data_selag,family="betabinomial",
                            weights=grazing_weights)
path1_mod4_bb<-glmmTMB(prop_pred_seeds~FFD_s_y+n_fl_s_y+(1|id)+(1|year),
                            data=data_selag,family="betabinomial",
                            weights=round(n_seeds))
```

Model diagnostics

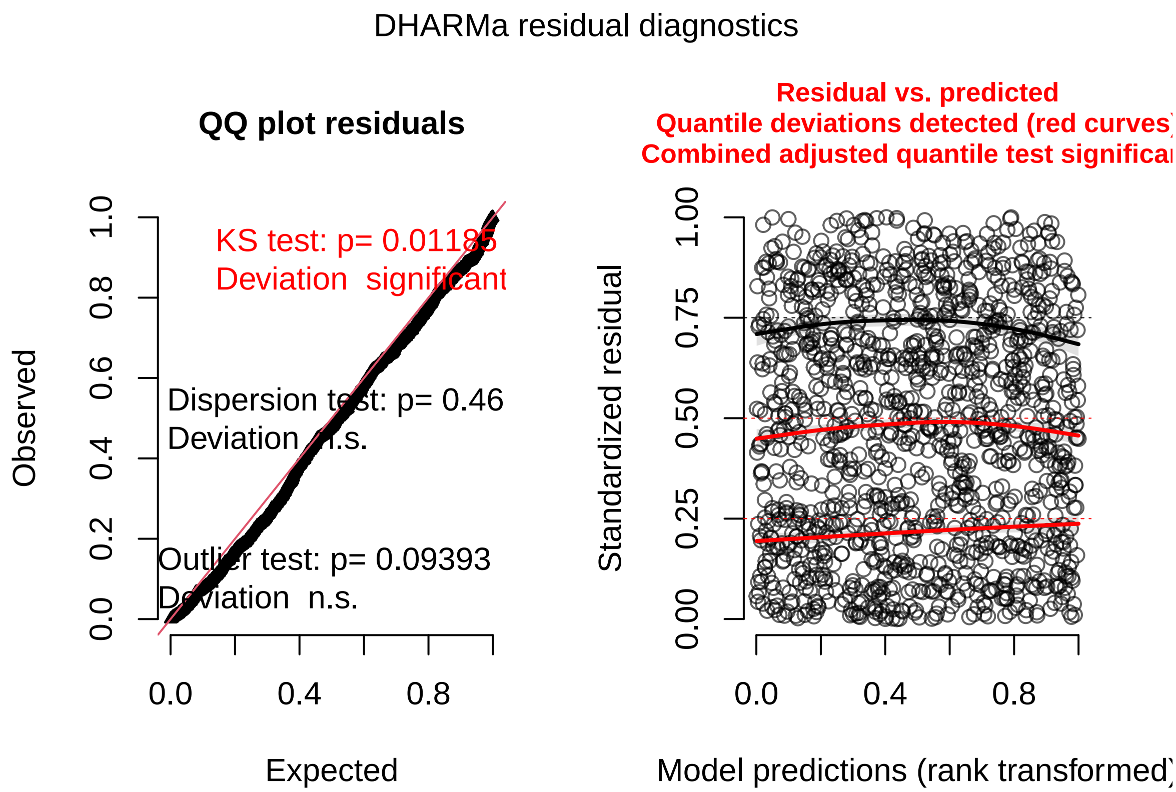
```
sim_path1_mod3_bb <- simulateResiduals(fittedModel = path1_mod3_bb, n = 5000)
sim_path1_mod4_bb <- simulateResiduals(fittedModel = path1_mod4_bb, n = 5000)
```

qq-plot and plot of residuals vs. predicted:

```
plot(sim_path1_mod3_bb,quantreg=T) # not OK
```




```
plot(sim_path1_mod4_bb,quantreg=T) # not OK
```

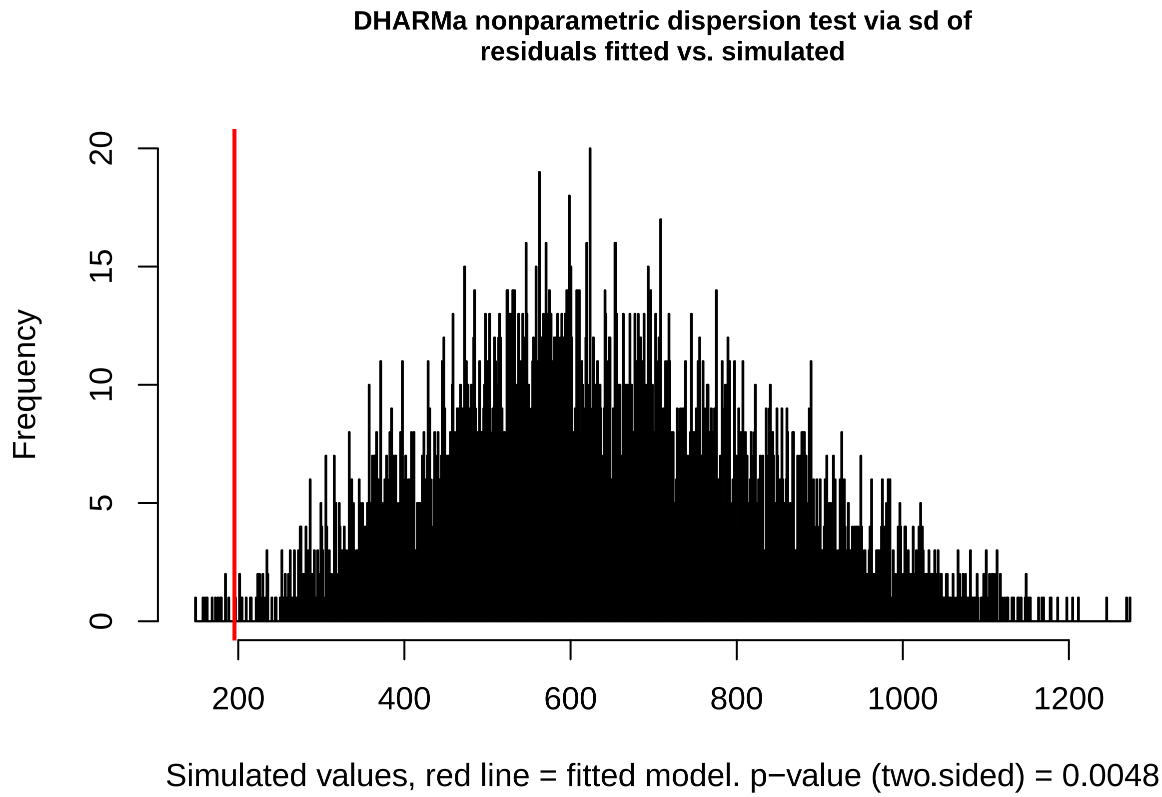


KS test being significant is probably not a problem.

From https://github.com/florianhartig/DHARMa/issues/181

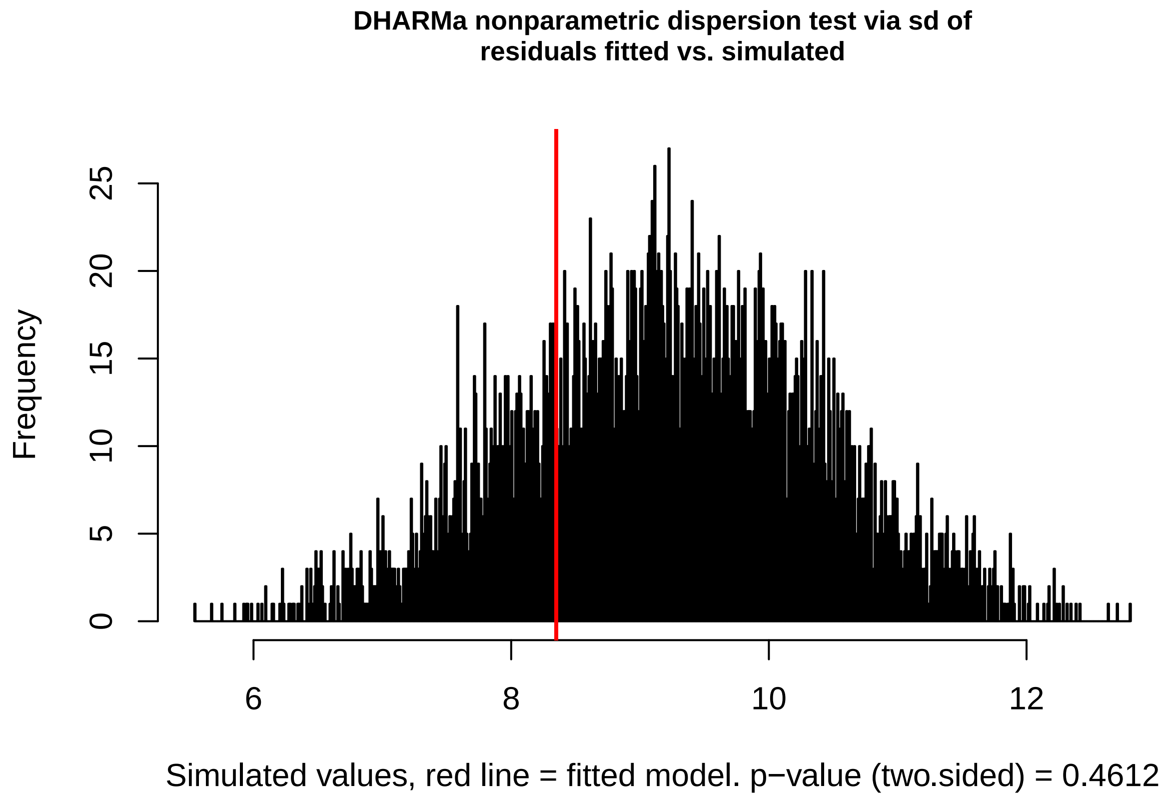
"the p-value shows you that there is a significant deviation from the assumed distribution, but significance != effect size. In other words, if you have a large number of data points (as you have here), even the slightest deviation will become significant. [...] the qq-plot is nearly linear, suggesting that the overall distribution is roughly OK.

```
testDispersion(sim_path1_mod3_bb) # a bit of underdispersion, probably OK
```



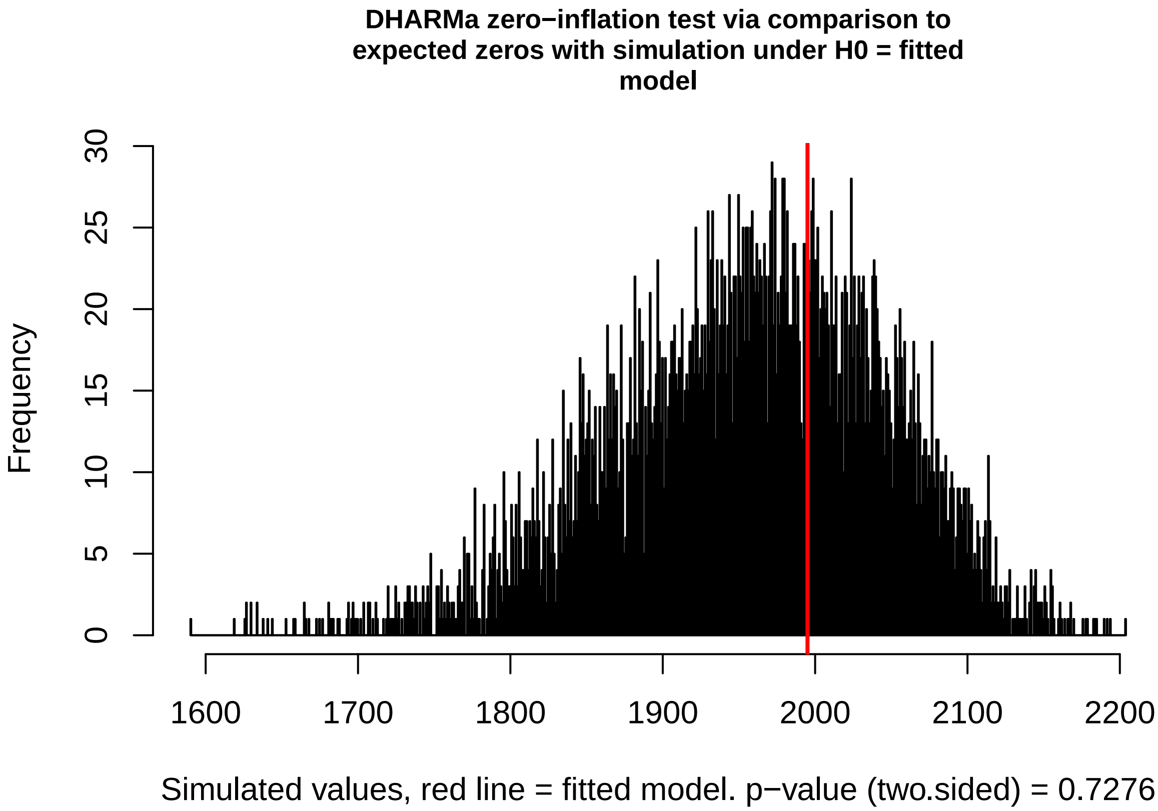

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.30572, p-value = 0.0048
## alternative hypothesis: two.sided
```

```
testDispersion(sim_path1_mod4_bb) # OK
```



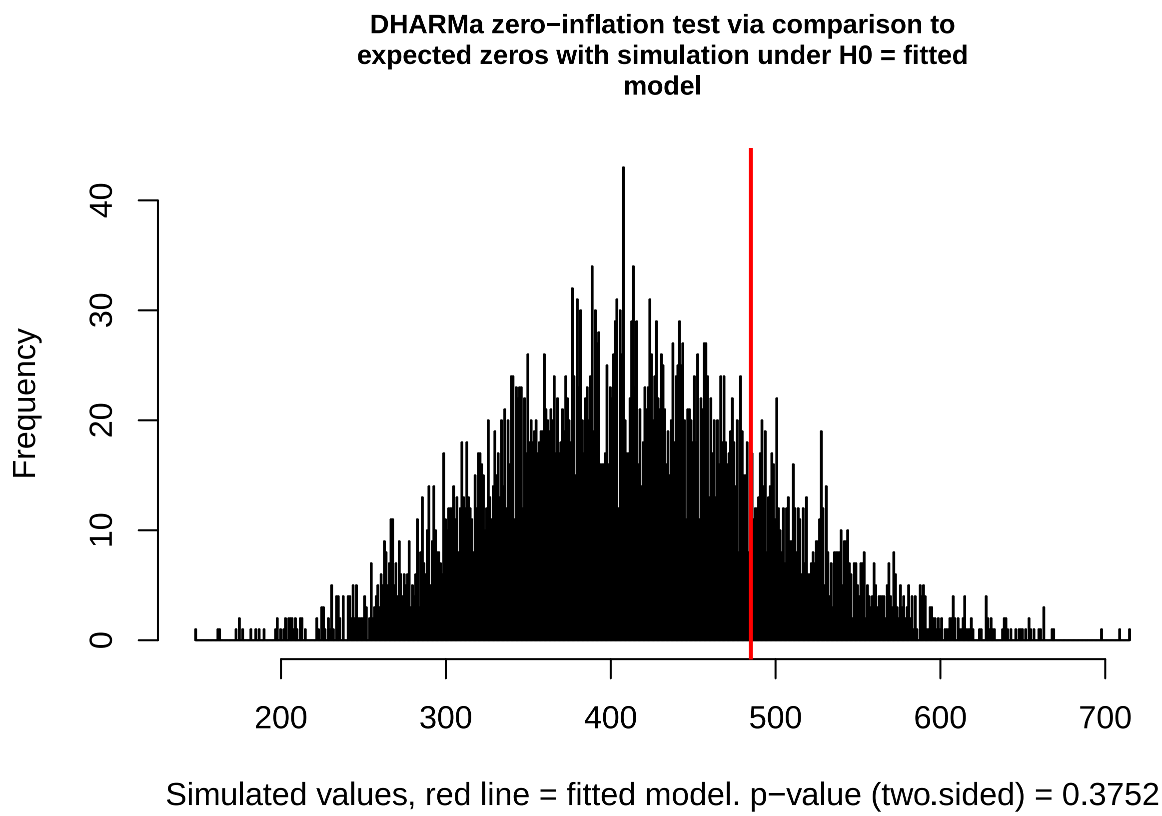

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.90432, p-value = 0.4612
## alternative hypothesis: two.sided
```

```
testZeroInflation(sim_path1_mod3_bb) # OK
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.0197, p-value = 0.7276
## alternative hypothesis: two.sided
```

```
testZeroInflation(sim_path1_mod4_bb) # OK
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.1793, p-value = 0.3752
## alternative hypothesis: two.sided
```

```
anova(path1_mod3,path1_mod3_bb)
```

```
## Data: data_selag
## Models:
## path1_mod3: grazing_corr ~ FFD_s_y + n_fl_s_y + (1 | id) + (1 | year), zi=~0, disp=~1
## path1_mod3_bb: grazing_corr ~ FFD_s_y + n_fl_s_y + (1 | id) + (1 | year), zi=~0, disp=~1
##               Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## path1_mod3     5 155103 155132 -77547   155093
## path1_mod3_bb  6   3601   3636  -1795     3589 151504      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(path1_mod4,path1_mod4_bb)
```

```
## Data: data_selag
## Models:
```

```
## path1_mod4: prop_pred_seeds ~ FFD_s_y + n_fl_s_y + (1 | id) + (1 | year), zi=~0, disp=~1
## path1_mod4_bb: prop_pred_seeds ~ FFD_s_y + n_fl_s_y + (1 | id) + (1 | year), zi=~0, disp=~1
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## path1_mod4      5 6696.0 6721.8 -3343.0   6686.0
## path1_mod4_bb   6 4953.5 4984.4 -2470.8   4941.5 1744.5      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(path1_mod3,path1_mod3_bb)
```

```
##                df        AIC
## path1_mod3      5 155103.361
## path1_mod3_bb   6   3601.166
```

```
AIC(path1_mod4,path1_mod4_bb)
```

```
##                df      AIC
## path1_mod4      5 6695.992
## path1_mod4_bb   6 4953.500
```

Use betabinomial models.

Tested quadratic effects of FFD on grazing and seed predation, and they were not significant in betabinomial models.

Boostrapped confidence intervals for model 2:

Mixed model

```
path1_mod2_glmer<-glmer(fitness_rel_y~FFD_s_y+n_fl_s_y+grazing_corr+
                          prop_pred_seeds+(1|id)+(1|year),data=data_selag)
# We need to use glmer for this
```

```
confint(b_par_1,level=0.95, method="boot") # Percentile method
```

```
##                      2.5 %      97.5 %
## (Intercept)      2.3664280  5.93437756
## FFD_s_y         -0.3156583 -0.04331491
## n_fl_s_y         0.4743067  0.69936623
## grazing_corr    -2.1922090 -0.53065021
## prop_pred_seeds -4.0396053 -3.24794402
```

Similar to significances in the lmer model.

LM (Elsa's code) (not used)

Similar to significances in the lmer model except for grazing, where the CIs overlap zero here. Use only percentile method?

New path model

```
summary(path1_mod1)$coefficients
```

```
## $cond
##                 Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)    0.5448470 0.21257583   2.563071 1.037508e-02
## FFD_s_y       -0.6968366 0.06747741 -10.326961 5.321988e-25
## n_fl_s_y       0.6111297 0.07683091   7.954216 1.802689e-15
## grazing_corr  -3.8250266 0.32225988 -11.869385 1.707182e-32
##
## $zi
## NULL
##
## $disp
## NULL
```

```
# std coefs if needed from path1 without random effects
summary(path1_mod2)$coefficients
```

```
## $cond
##                   Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)      4.1866333 0.88694757   4.720271 2.355301e-06
## FFD_s_y         -0.1790698 0.06873317  -2.605290 9.179656e-03
## n_fl_s_y         0.5868972 0.05766612  10.177505 2.498870e-24
## grazing_corr    -1.3597652 0.42561090  -3.194855 1.399010e-03
## prop_pred_seeds -3.6442115 0.20386754 -17.875388 1.833937e-71
##
## $zi
## NULL
##
## $disp
## NULL
```

```
# get significances from bootstrapped CIs, std coefs from path1 without random effects
summary(path1_mod3_bb)$coefficients
```

```
## $cond
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -2.9368887 0.32909846 -8.924043 4.495707e-19
## FFD_s_y     -0.3479031 0.08335881 -4.173561 2.998753e-05
## n_fl_s_y     0.1875620 0.06457848  2.904404 3.679536e-03
##
## $zi
## NULL
##
## $disp
## NULL
```

```r
# significances and ustd coefs from here, std if needed from path1 without random effects
summary(path1_mod4_bb)$coefficients
```

```
## $cond
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -0.7491371 0.31175650 -2.402956 1.626315e-02
## FFD_s_y      0.1007740 0.05463502  1.844495 6.511103e-02
## n_fl_s_y     0.2060346 0.04261941  4.834291 1.336212e-06
##
## $zi
## NULL
##
## $disp
## NULL
```

```r
# significances and ustd coefs from here, std if needed from path1 without random effects
```

piecewiseSEM does not work with glmmTMB - and that is the only package I found that fits betabinomial mixed models.

```r
path1_norandom<-psem(
  glm(seeds_01~FFD_s_y+n_fl_s_y+grazing_corr,
               data=data_selag,family="binomial"),
  lm(fitness_rel_y~FFD_s_y+n_fl_s_y+grazing_corr+prop_pred_seeds,
              data=data_selag),
  glm(grazing_corr~FFD_s_y+n_fl_s_y,
                 data=data_selag,family="binomial",weights=grazing_weights),
  glm(prop_pred_seeds~FFD_s_y+n_fl_s_y,
                 data=data_selag,family="binomial",
                 weights=n_seeds))
coefs(path1_norandom)
```

```
##           Response       Predictor Estimate Std.Error   DF Crit.Value P.Value
## 1          seeds_01         FFD_s_y  -0.6041    0.0559 2372   -10.8057  0.0000
## 2          seeds_01        n_fl_s_y   0.4999    0.0642 2372     7.7907  0.0000
## 3          seeds_01    grazing_corr  -3.2497    0.2395 2372   -13.5712  0.0000
## 4     fitness_rel_y         FFD_s_y  -0.2054    0.0881 1276    -2.3322  0.0198
## 5     fitness_rel_y        n_fl_s_y   0.4506    0.0726 1276     6.2070  0.0000
## 6     fitness_rel_y    grazing_corr   0.6428    0.5272 1276     1.2192  0.2230
## 7     fitness_rel_y prop_pred_seeds  -1.6647    0.1931 1276    -8.6187  0.0000
## 8      grazing_corr         FFD_s_y  -0.1169    0.0054 2350   -21.5365  0.0000
## 9      grazing_corr        n_fl_s_y  -0.0692    0.0033 2350   -20.7038  0.0000
## 10 prop_pred_seeds         FFD_s_y  -0.0330    0.0208 1278    -1.5837  0.1133
## 11 prop_pred_seeds        n_fl_s_y   0.0893    0.0120 1278     7.4500  0.0000
##    Std.Estimate
## 1       -0.2741 ***
## 2        0.2268 ***
## 3       -0.4572 ***
## 4       -0.0965   *
## 5        0.2118 ***
## 6        0.0937
## 7       -0.2859 ***
```

```
## 8        -0.0641 ***
## 9        -0.0379 ***
## 10       -0.0181
## 11        0.0490 ***
```

```
plot(path1_norandom)
```

Figure 1: Path diagram (made in Inkscape)

% of grazed plants that produced seeds

```
nrow(subset(data_selag,grazing_corr>0&n_seeds>0))*100/
  nrow(subset(data_selag,grazing_corr>0))
```

```
## [1] 28.63341
```

% of plants with more than half of their above-ground structures removed that produced seeds

```
nrow(subset(data_selag,grazing_corr>0.5&n_seeds>0))*100/
  nrow(subset(data_selag,grazing_corr>0))
```

```
## [1] 6.290672
```

Only 29% of plants that experienced any grazing produced seeds, and 6% of plants that had more than half of their above-ground structures removed.

H3

H3: Variation in climatic conditions during spring among years causes differences in selection on flowering time, through effects mediated by the intensity of biotic interactions.

H3 - Part 1

Variation in climatic conditions during spring among years influences the intensity of biotic interactions and the covariance between interaction intensity and plant phenology.

First, models to test the effect of spring temperatures on antagonistic interactions, while accounting for FFD and number of flowers. Using FFD and number of flowers standardized within years, temperatures standardized across years. Including interactions among FFD and temperatures. NOT including a random effect of year because there is only one value of temperature for each year.

Grazing

```
mod_grazing<-glmmTMB(cbind(grazing_success,grazing_failure)~
                      (scale(mean_3)+scale(mean_4)+scale(mean_5))*FFD_s_y+
                      n_fl_s_y+(1|id),data = data_selag,
                    family="binomial")
summary(mod_grazing)
```

```
##  Family: binomial  ( logit )
## Formula:
## cbind(grazing_success, grazing_failure) ~ (scale(mean_3) + scale(mean_4) +
##     scale(mean_5)) * FFD_s_y + n_fl_s_y + (1 | id)
## Data: data_selag
##
##      AIC      BIC   logLik deviance df.resid
## 174950.7 175008.3 -87465.4 174930.7     2344
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 48.66    6.976
## Number of obs: 2354, groups:  id, 834
##
## Conditional model:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -8.990310   0.459323  -19.57  < 2e-16 ***
## scale(mean_3)          0.683845   0.007932   86.21  < 2e-16 ***
## scale(mean_4)          0.022606   0.005433    4.16 3.17e-05 ***
## scale(mean_5)         -0.066343   0.028336   -2.34   0.0192 *
## FFD_s_y               -0.942016   0.015389  -61.22  < 2e-16 ***
## n_fl_s_y              -0.182051   0.005101  -35.69  < 2e-16 ***
## scale(mean_3):FFD_s_y  0.522687   0.008510   61.42  < 2e-16 ***
## scale(mean_4):FFD_s_y  0.168588   0.005916   28.50  < 2e-16 ***
## scale(mean_5):FFD_s_y  1.788498   0.031905   56.06  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

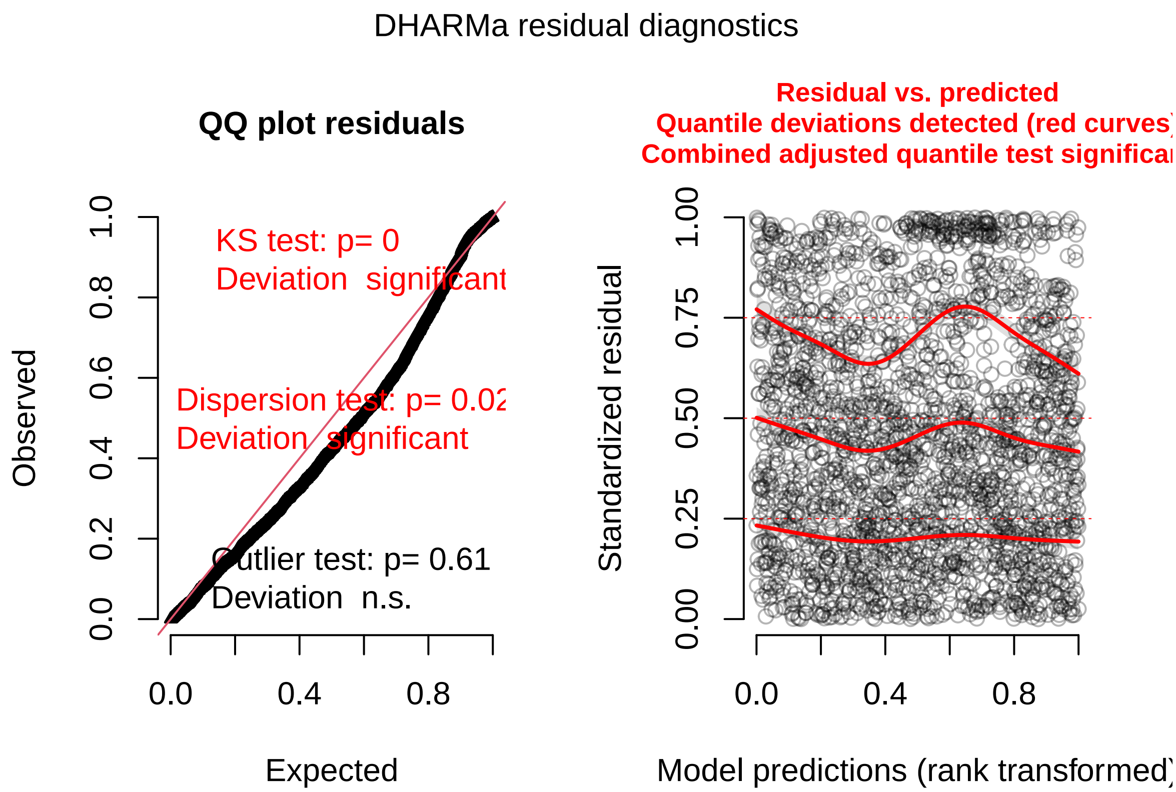
Model diagnostics

Simulate residuals of the model:

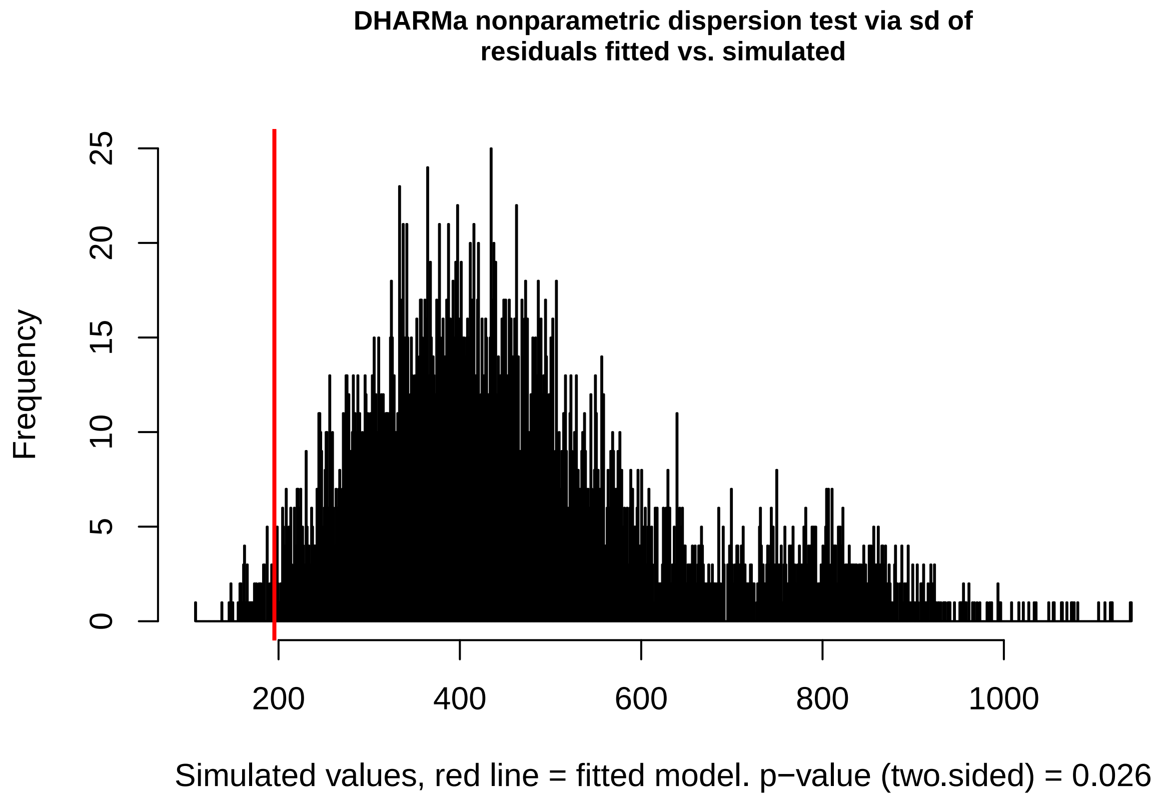
```
sim_mod_grazing <- simulateResiduals(fittedModel=mod_grazing,n=5000)
```

qq-plot and plot of residuals vs. predicted:

```
plot(sim_mod_grazing,quantreg=T)
```



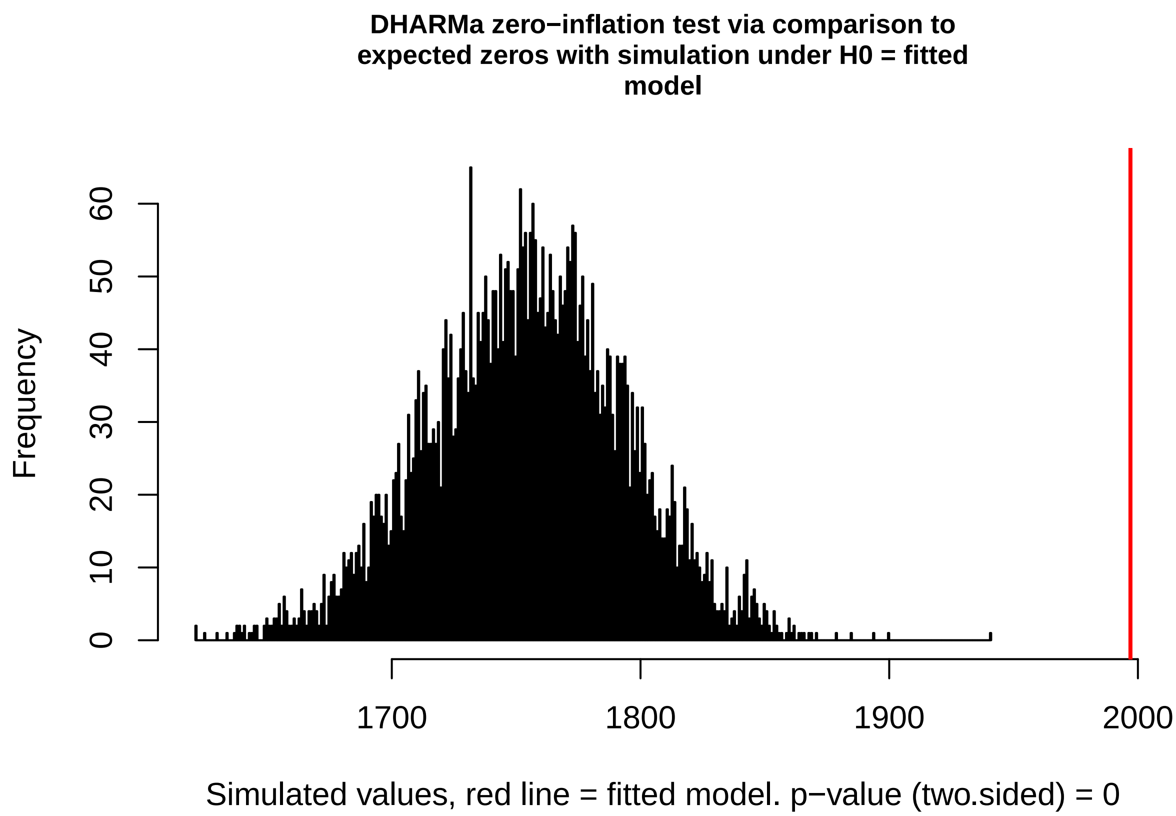

```
testDispersion(sim_mod_grazing)
```




```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.41949, p-value = 0.026
## alternative hypothesis: two.sided
```

Slightly underdispersed.

```
testZeroInflation(sim_mod_grazing)
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.1381, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Test of zero inflation is significant.

Try with betabinomial

```
mod_grazing_bb<-glmmTMB(cbind(grazing_success,grazing_failure)~
                        (scale(mean_3)+scale(mean_4)+scale(mean_5))*FFD_s_y+
                        n_fl_s_y+(1|id),data = data_selag,
                      family="betabinomial")
summary(mod_grazing_bb)
```

```
##  Family: betabinomial  ( logit )
## Formula:
## cbind(grazing_success, grazing_failure) ~ (scale(mean_3) + scale(mean_4) +
##     scale(mean_5)) * FFD_s_y + n_fl_s_y + (1 | id)
## Data: data_selag
```

```
##
##      AIC      BIC   logLik deviance df.resid
##   4013.7   4077.1  -1995.8   3991.7     2343
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.2367   0.4865
## Number of obs: 2354, groups:  id, 834
##
## Overdispersion parameter for betabinomial family (): 0.141
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.23190    0.09050 -24.663  < 2e-16 ***
## scale(mean_3)          0.50384    0.06704   7.516 5.66e-14 ***
## scale(mean_4)         -0.33282    0.06416  -5.187 2.14e-07 ***
## scale(mean_5)         -0.44010    0.08623  -5.104 3.32e-07 ***
## FFD_s_y               -0.25577    0.07307  -3.500 0.000464 ***
## n_fl_s_y               0.15921    0.05901   2.698 0.006976 **
## scale(mean_3):FFD_s_y  0.10707    0.07179   1.491 0.135832
## scale(mean_4):FFD_s_y -0.12757    0.06553  -1.947 0.051571 .
## scale(mean_5):FFD_s_y  0.02182    0.08916   0.245 0.806639
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

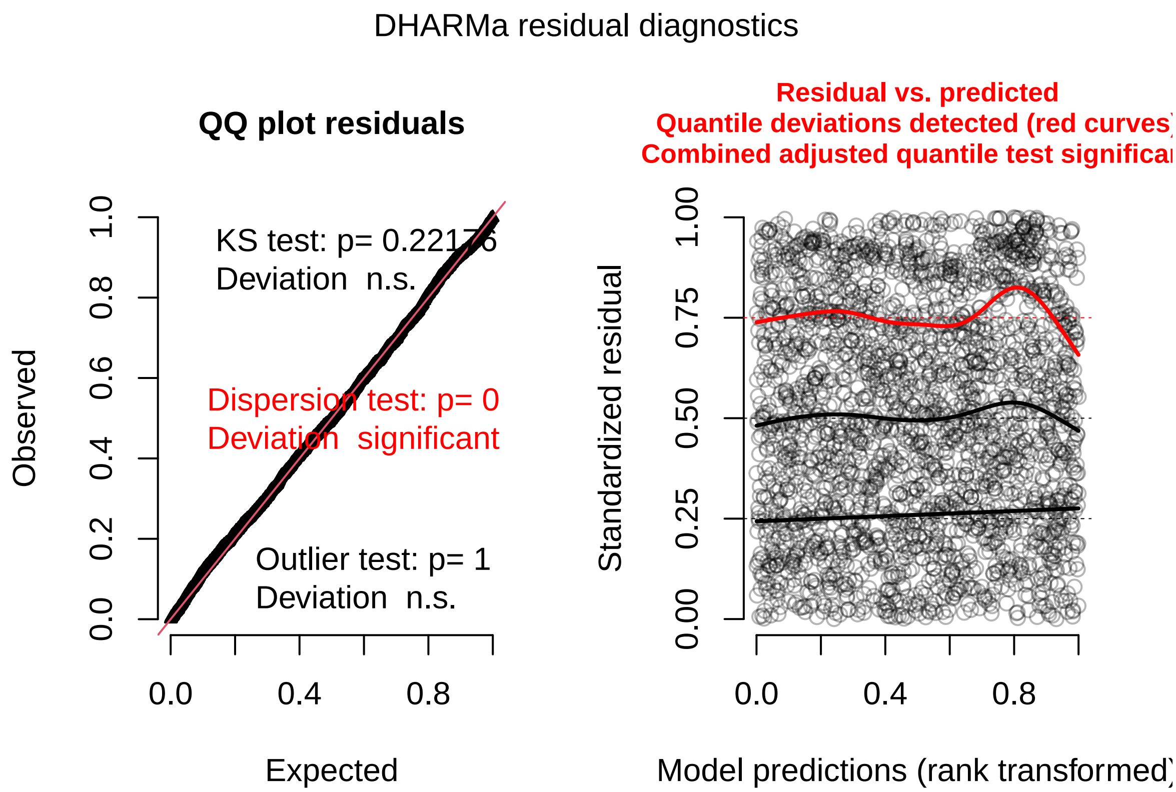
Model diagnostics

Simulate residuals of the model:

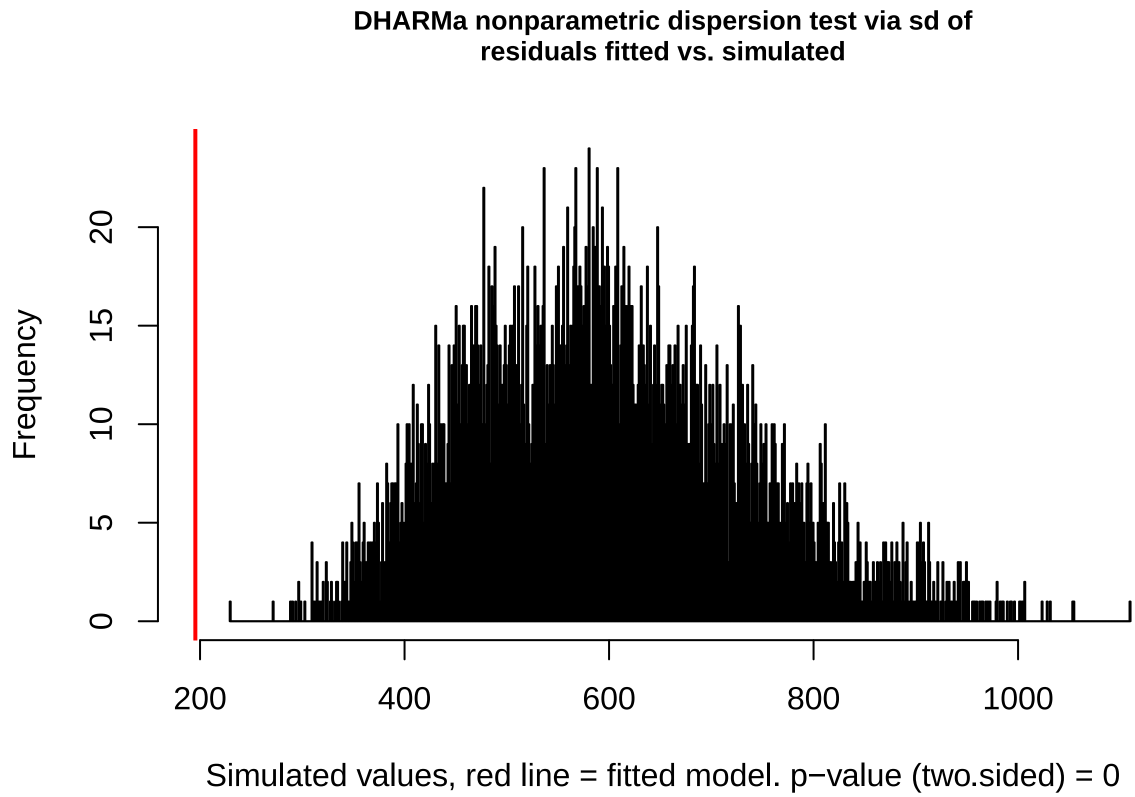
```
sim_mod_grazing_bb <- simulateResiduals(fittedModel=mod_grazing_bb,n=5000)
```

qq-plot and plot of residuals vs. predicted:

```
plot(sim_mod_grazing_bb,quantreg=T)
```



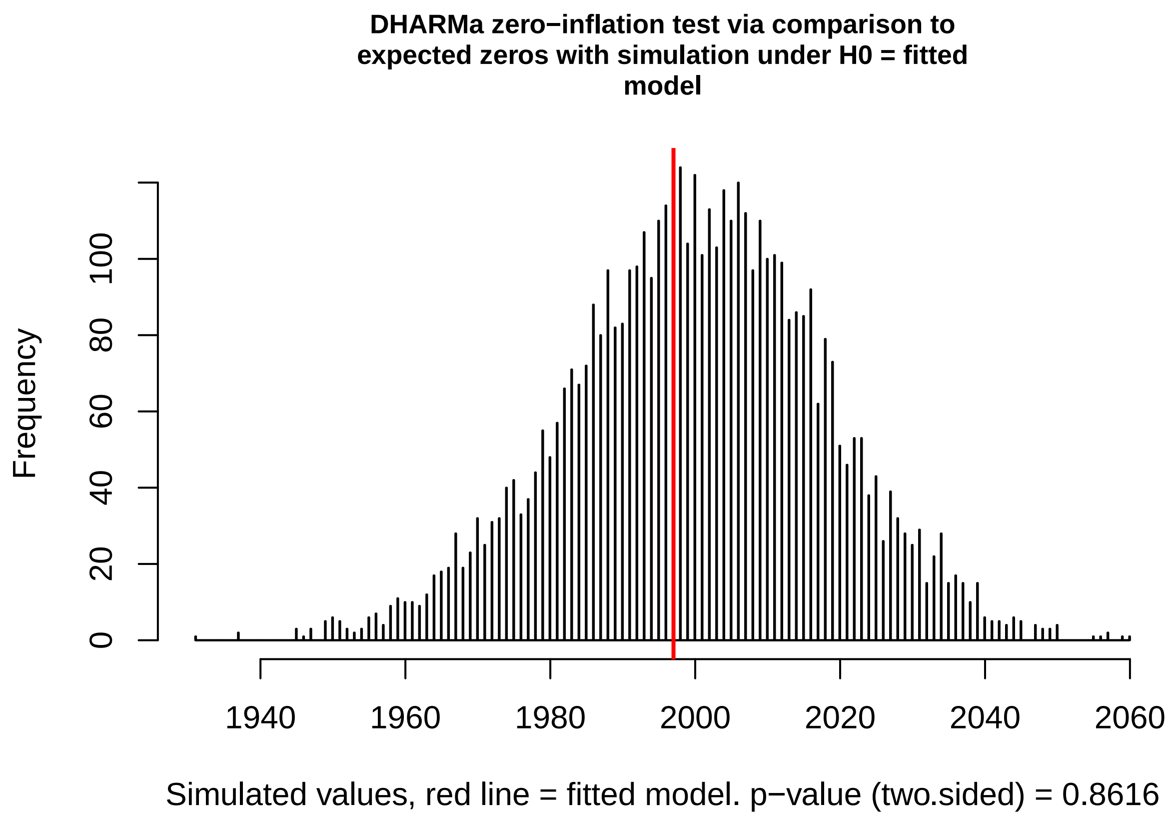

```
testDispersion(sim_mod_grazing_bb)
```




```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.32727, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Still underdispersion.

```r
testZeroInflation(sim_mod_grazing_bb)
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 0.9984, p-value = 0.8616
## alternative hypothesis: two.sided
```

But zero inflation is fixed

```
anova(mod_grazing,mod_grazing_bb)
```

```
## Data: data_selag
## Models:
## mod_grazing: cbind(grazing_success, grazing_failure) ~ (scale(mean_3) + scale(mean_4) + , zi=~0, disp
## mod_grazing:     scale(mean_5)) * FFD_s_y + n_fl_s_y + (1 | id), zi=~0, disp=~1
## mod_grazing_bb: cbind(grazing_success, grazing_failure) ~ (scale(mean_3) + scale(mean_4) + , zi=~0, d
## mod_grazing_bb:     scale(mean_5)) * FFD_s_y + n_fl_s_y + (1 | id), zi=~0, disp=~1
##                Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_grazing    10 174951 175008 -87465   174931
## mod_grazing_bb 11   4014   4077  -1996     3992 170939      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(mod_grazing,mod_grazing_bb)
```

```
##                df        AIC
## mod_grazing    10 174950.700
## mod_grazing_bb 11   4013.658
```

Keep mod_grazing_bb.

Seed predation

```r
mod_seedpred<-glmmTMB(cbind(round(n_pred_seeds),round(n_intact_seeds))~
                      (scale(mean_3)+scale(mean_4)+scale(mean_5)+scale(mean_6))*
                       FFD_s_y+n_fl_s_y+(1|id),data=subset(data_selag,n_seeds>0),
                     family="binomial")
summary(mod_seedpred)
```

```
##  Family: binomial  ( logit )
## Formula:
## cbind(round(n_pred_seeds), round(n_intact_seeds)) ~ (scale(mean_3) +
##     scale(mean_4) + scale(mean_5) + scale(mean_6)) * FFD_s_y +
##     n_fl_s_y + (1 | id)
## Data: subset(data_selag, n_seeds > 0)
##
##      AIC      BIC   logLik deviance df.resid
##   9018.9   9080.7  -4497.4   8994.9     1269
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 2.281    1.51
## Number of obs: 1281, groups:  id, 593
##
## Conditional model:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.41142    0.07781 -18.138  < 2e-16 ***
## scale(mean_3)         -0.13644    0.03757  -3.632 0.000282 ***
## scale(mean_4)         -0.25152    0.03480  -7.228 4.90e-13 ***
## scale(mean_5)          0.21985    0.04841   4.541 5.59e-06 ***
## scale(mean_6)          0.55555    0.03961  14.026  < 2e-16 ***
## FFD_s_y                0.00579    0.03651   0.159 0.873979
## n_fl_s_y               0.07319    0.02312   3.165 0.001550 **
## scale(mean_3):FFD_s_y  0.38384    0.04201   9.136  < 2e-16 ***
## scale(mean_4):FFD_s_y -0.34184    0.03692  -9.258  < 2e-16 ***
## scale(mean_5):FFD_s_y -0.03326    0.05404  -0.615 0.538252
## scale(mean_6):FFD_s_y -0.06563    0.04443  -1.477 0.139633
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

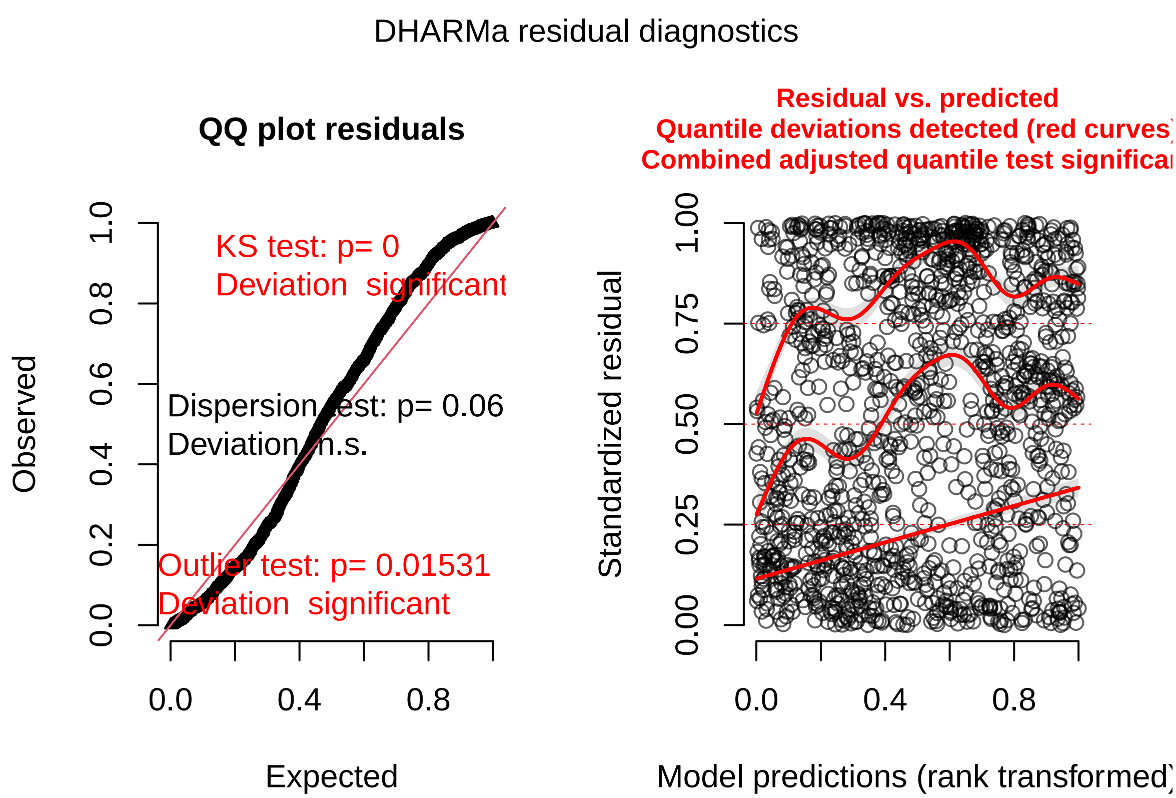
Model diagnostics

Simulate residuals of the model:

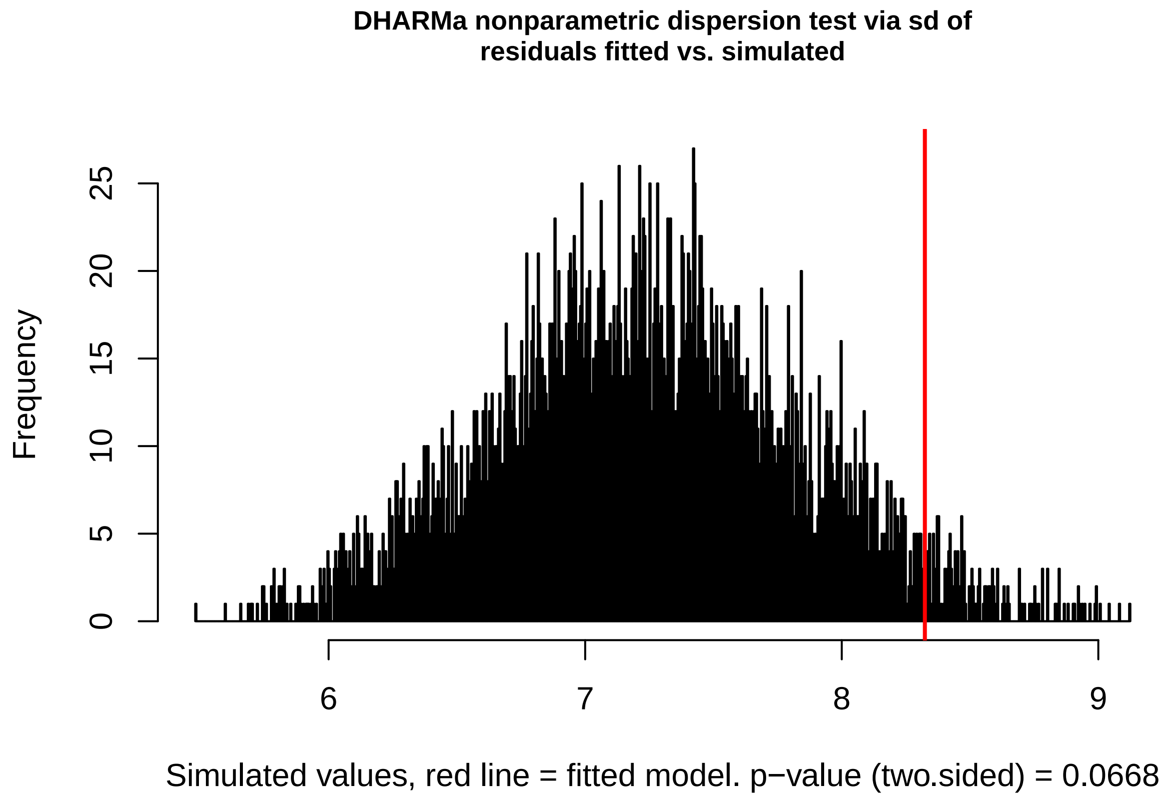
```
sim_mod_seedpred <- simulateResiduals(fittedModel=mod_seedpred,n=5000)
```

qq-plot and plot of residuals vs. predicted:

```
plot(sim_mod_seedpred,quantreg=T)
```



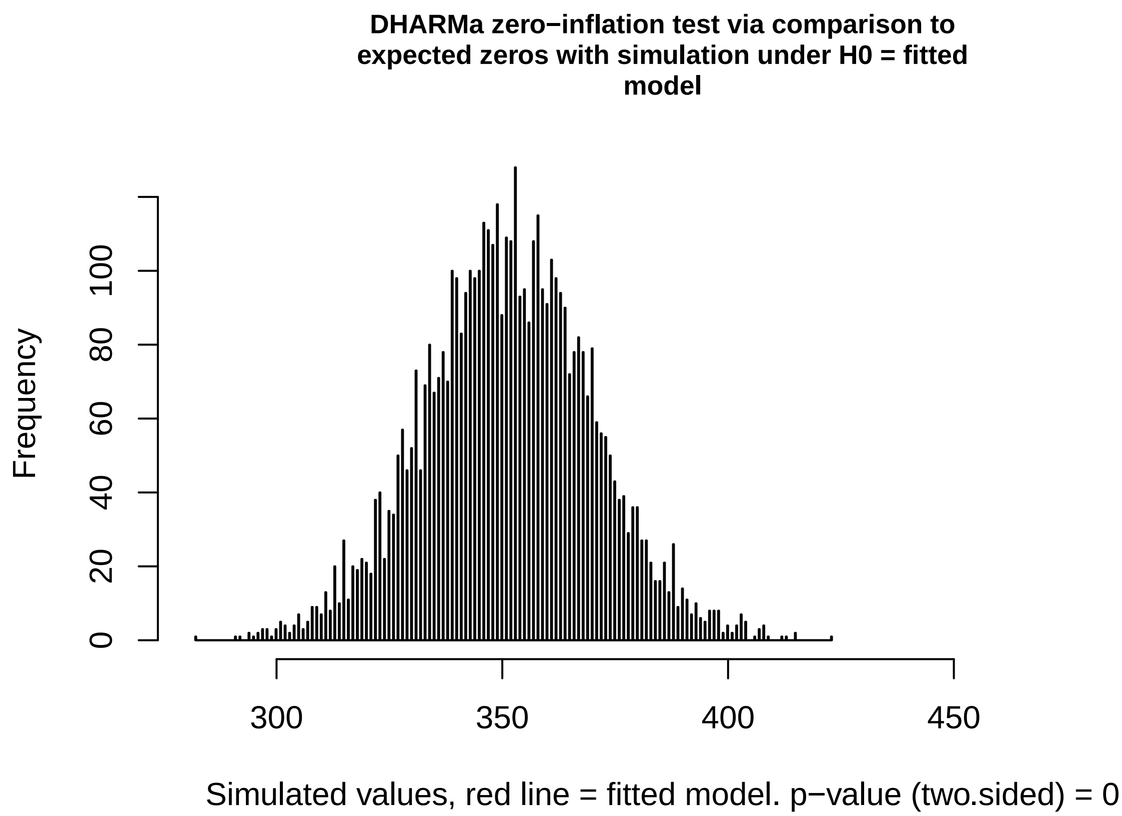

```
testDispersion(sim_mod_seedpred)
```




```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 1.1495, p-value = 0.0668
## alternative hypothesis: two.sided
```

OK.

```
testZeroInflation(sim_mod_seedpred)
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.3914, p-value < 2.2e-16
## alternative hypothesis: two.sided
```

Zero inflation.

Try with betabinomial

```
mod_seedpred_bb<-glmmTMB(cbind(round(n_pred_seeds),round(n_intact_seeds))~
                      (scale(mean_3)+scale(mean_4)+scale(mean_5)+scale(mean_6))*
                       FFD_s_y+n_fl_s_y+(1|id),data=subset(data_selag,n_seeds>0),
                     family="betabinomial")
summary(mod_seedpred_bb)
```

```
##  Family: betabinomial  ( logit )
## Formula:
## cbind(round(n_pred_seeds), round(n_intact_seeds)) ~ (scale(mean_3) +
##     scale(mean_4) + scale(mean_5) + scale(mean_6)) * FFD_s_y +
##     n_fl_s_y + (1 | id)
```

```
## Data: subset(data_selag, n_seeds > 0)
##
##      AIC      BIC   logLik deviance df.resid
##   5339.2   5406.3  -2656.6   5313.2     1268
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev.
##  id     (Intercept) 0.06597  0.2569
## Number of obs: 1281, groups:  id, 593
##
## Overdispersion parameter for betabinomial family (): 1.15
##
## Conditional model:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.84346    0.05517 -15.289  < 2e-16 ***
## scale(mean_3)       -0.20163    0.07223  -2.792  0.00525 **
## scale(mean_4)        0.02973    0.06213   0.478  0.63233
## scale(mean_5)        0.46942    0.07236   6.487 8.75e-11 ***
## scale(mean_6)        0.41522    0.06211   6.685 2.31e-11 ***
## FFD_s_y              0.03140    0.05865   0.535  0.59231
## n_fl_s_y             0.17791    0.04409   4.035 5.46e-05 ***
## scale(mean_3):FFD_s_y  0.17398    0.08155   2.133  0.03289 *
## scale(mean_4):FFD_s_y -0.15858    0.06662  -2.380  0.01729 *
## scale(mean_5):FFD_s_y  0.02516    0.07756   0.324  0.74566
## scale(mean_6):FFD_s_y -0.09364    0.06820  -1.373  0.16972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

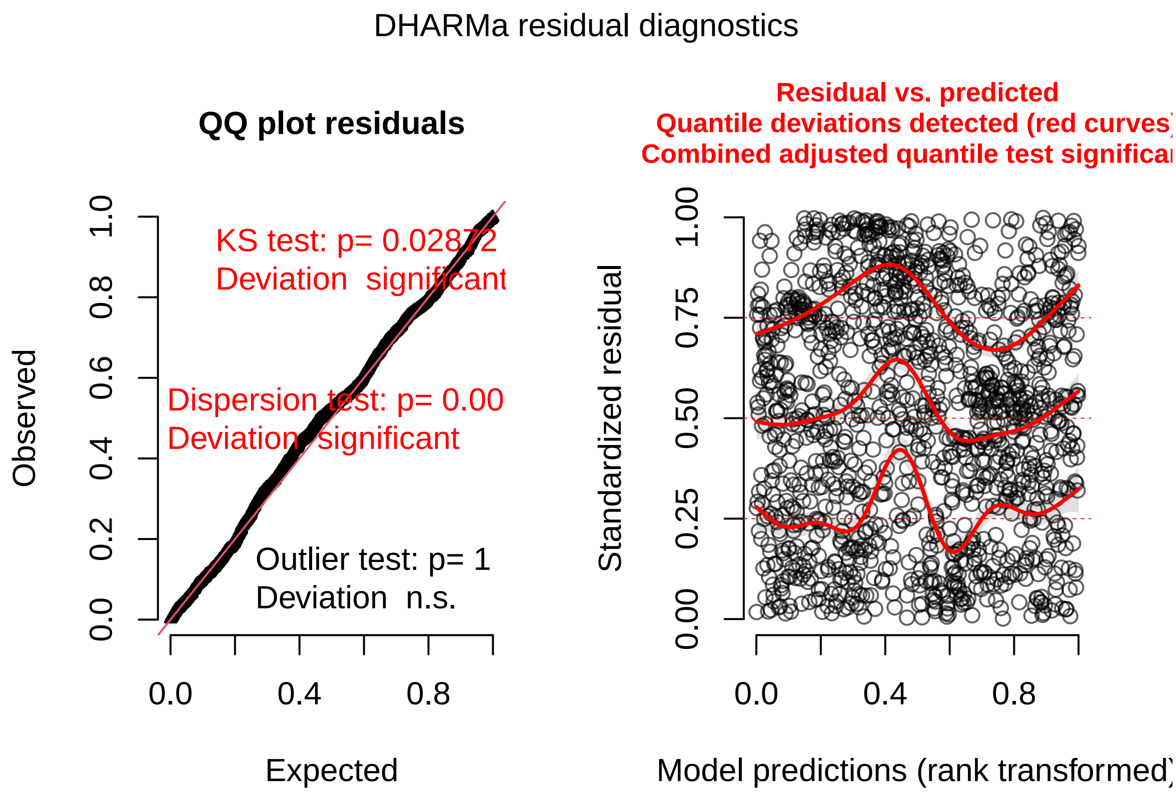
Model diagnostics

Simulate residuals of the model:

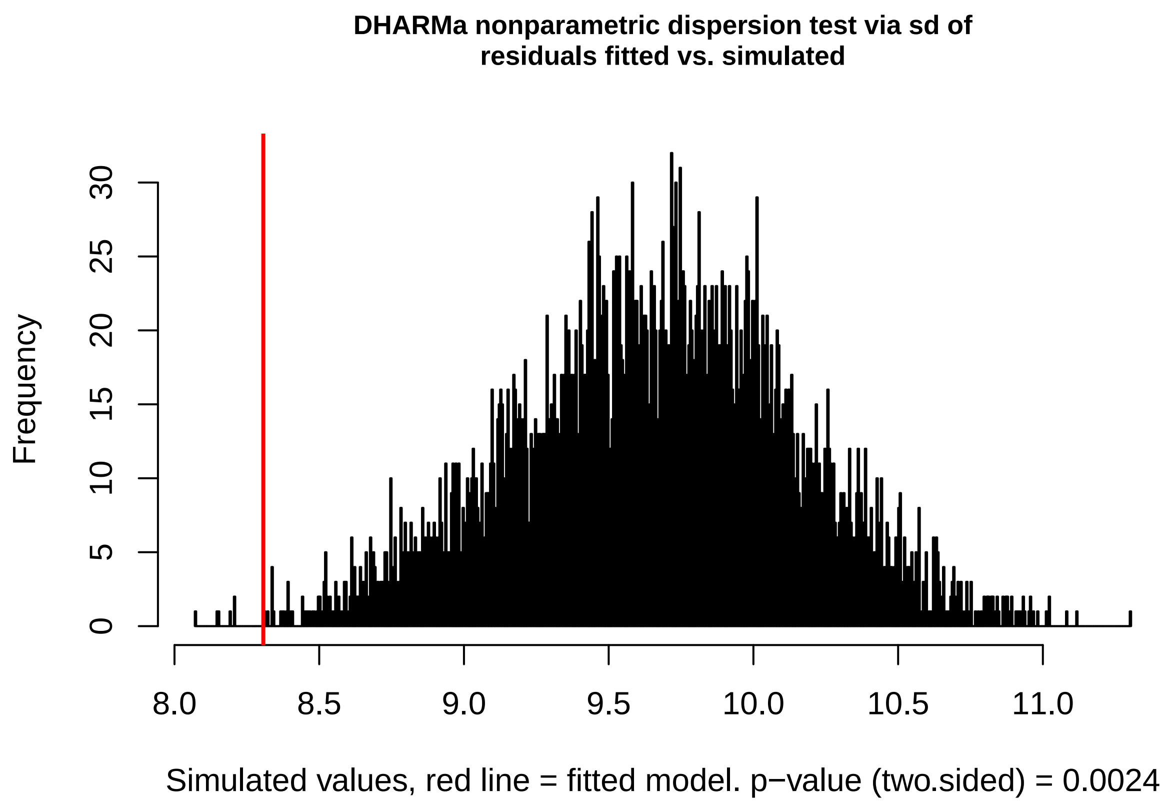
```
sim_mod_seedpred_bb <- simulateResiduals(fittedModel=mod_seedpred_bb,n=5000)
```

qq-plot and plot of residuals vs. predicted:

```
plot(sim_mod_seedpred_bb,quantreg=T)
```



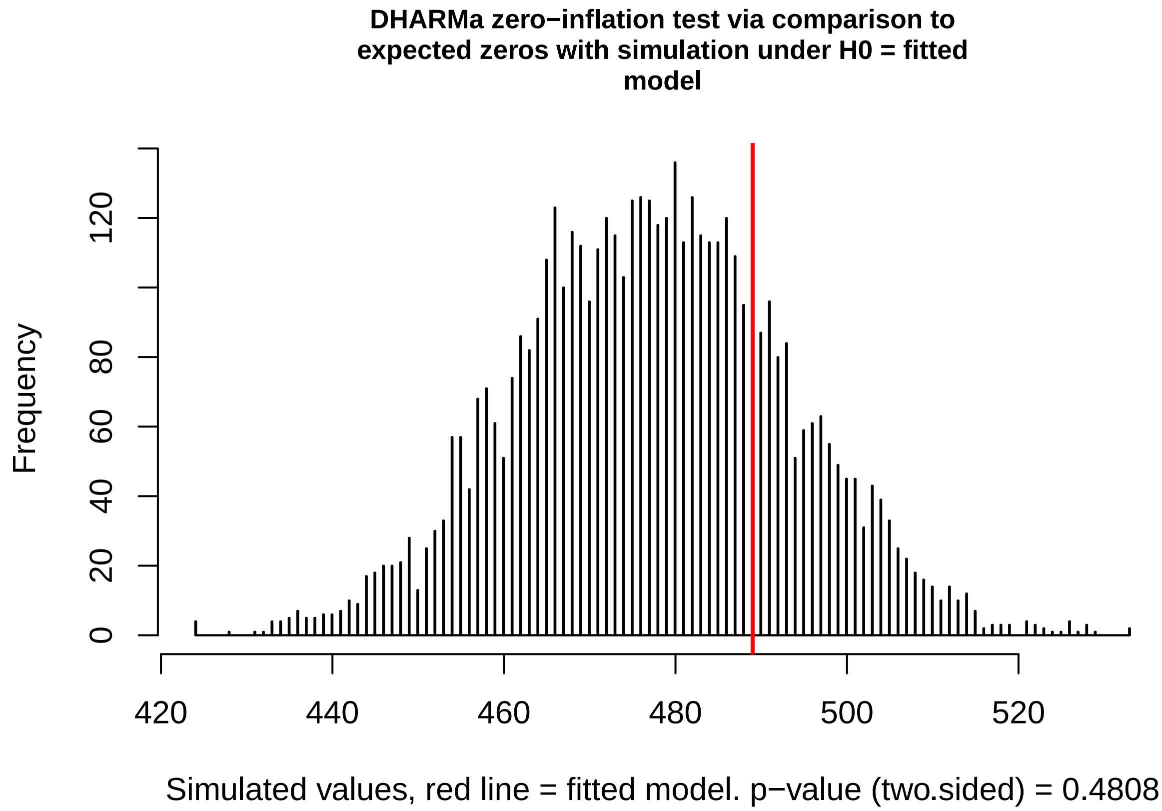

```r
testDispersion(sim_mod_seedpred_bb)
```




```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## ratioObsSim = 0.85972, p-value = 0.0024
## alternative hypothesis: two.sided
```

Slight underdispersion.

```
testZeroInflation(sim_mod_seedpred_bb)
```




```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 1.0244, p-value = 0.4808
## alternative hypothesis: two.sided
```

OK.

```
anova(mod_seedpred,mod_seedpred_bb)
```

```
## Data: subset(data_selag, n_seeds > 0)
## Models:
## mod_seedpred: cbind(round(n_pred_seeds), round(n_intact_seeds)) ~ (scale(mean_3) + , zi=~0, disp=~1
## mod_seedpred:     scale(mean_4) + scale(mean_5) + scale(mean_6)) * FFD_s_y + , zi=~0, disp=~1
## mod_seedpred:     n_fl_s_y + (1 | id), zi=~0, disp=~1
## mod_seedpred_bb: cbind(round(n_pred_seeds), round(n_intact_seeds)) ~ (scale(mean_3) + , zi=~0, disp=
## mod_seedpred_bb:     scale(mean_4) + scale(mean_5) + scale(mean_6)) * FFD_s_y + , zi=~0, disp=~1
## mod_seedpred_bb:     n_fl_s_y + (1 | id), zi=~0, disp=~1
##                 Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## mod_seedpred    12 9018.9 9080.7 -4497.4   8994.9
## mod_seedpred_bb 13 5339.2 5406.3 -2656.6   5313.2 3681.6      1  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```