Reworked analyses and figure drafts for paper1 after JEB reviews

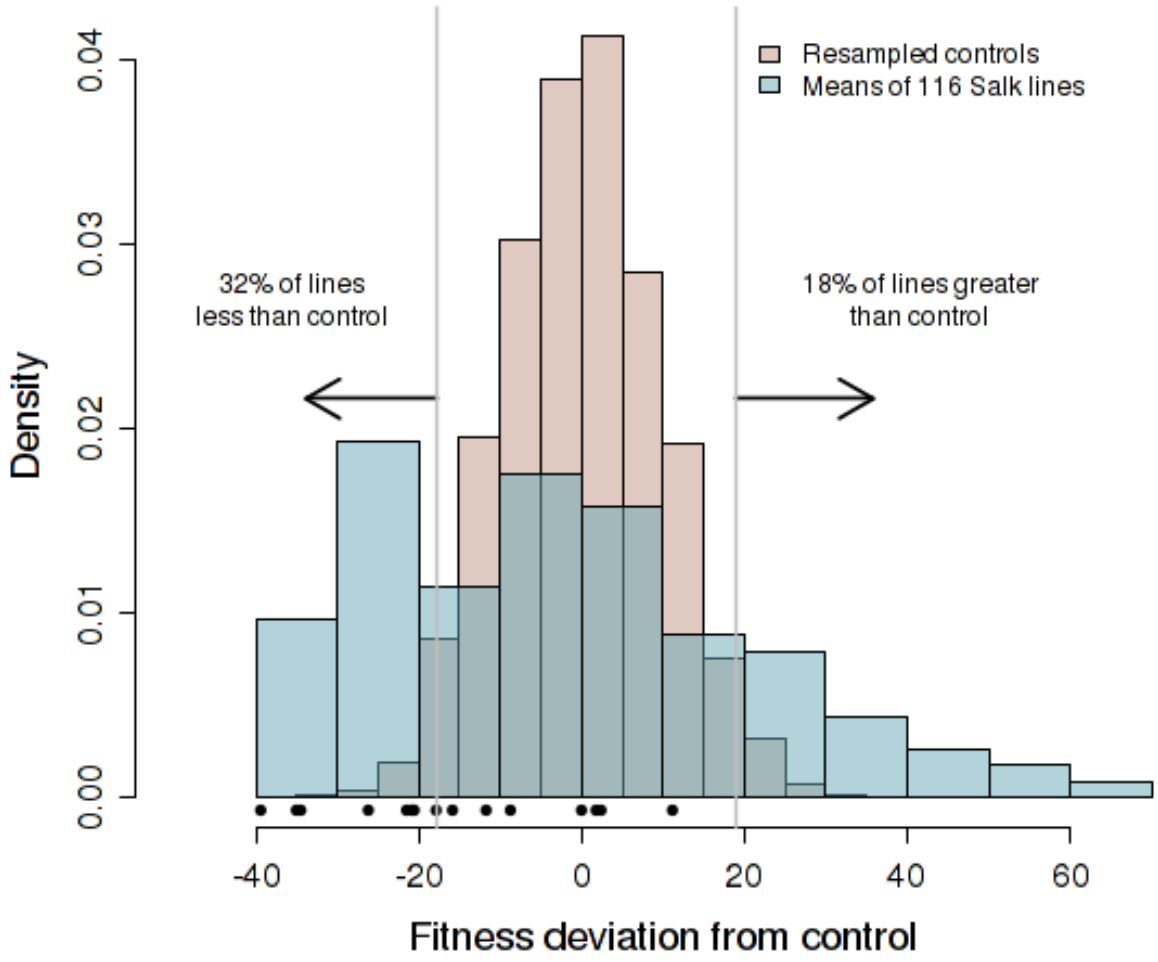
```
[1] "does gls exist TRUE"
```

1 Distribution of mutations

1.1 Distribution of mutations relative to control using means only.

Here are various figures emphasizing variation among mutant lines compared to controls. In this first figure, the distribution of line means is plotted with the distribution of all control plants as well as 116 resampled means of control replicates equal in size to the average number of reps per SALK line

This figure focuses on the resampled control means and compares to individual line means. This is the figure we placed in the ms



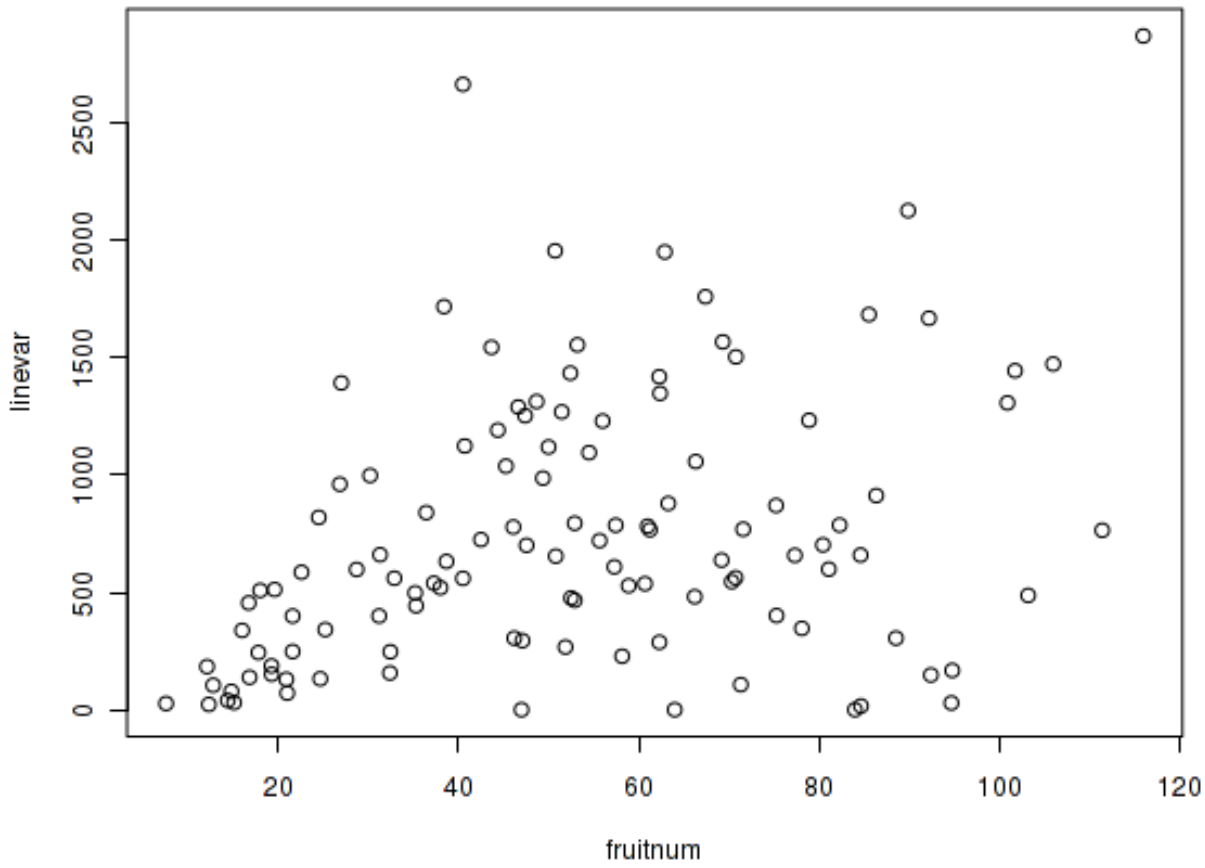
The next set of figures shows what happens when only small numbers of individuals are pulled from the full distribution. The sample sizes are: 2, 4, 8, and 10.

1.2 Additions suggested by JEB reviewer 1

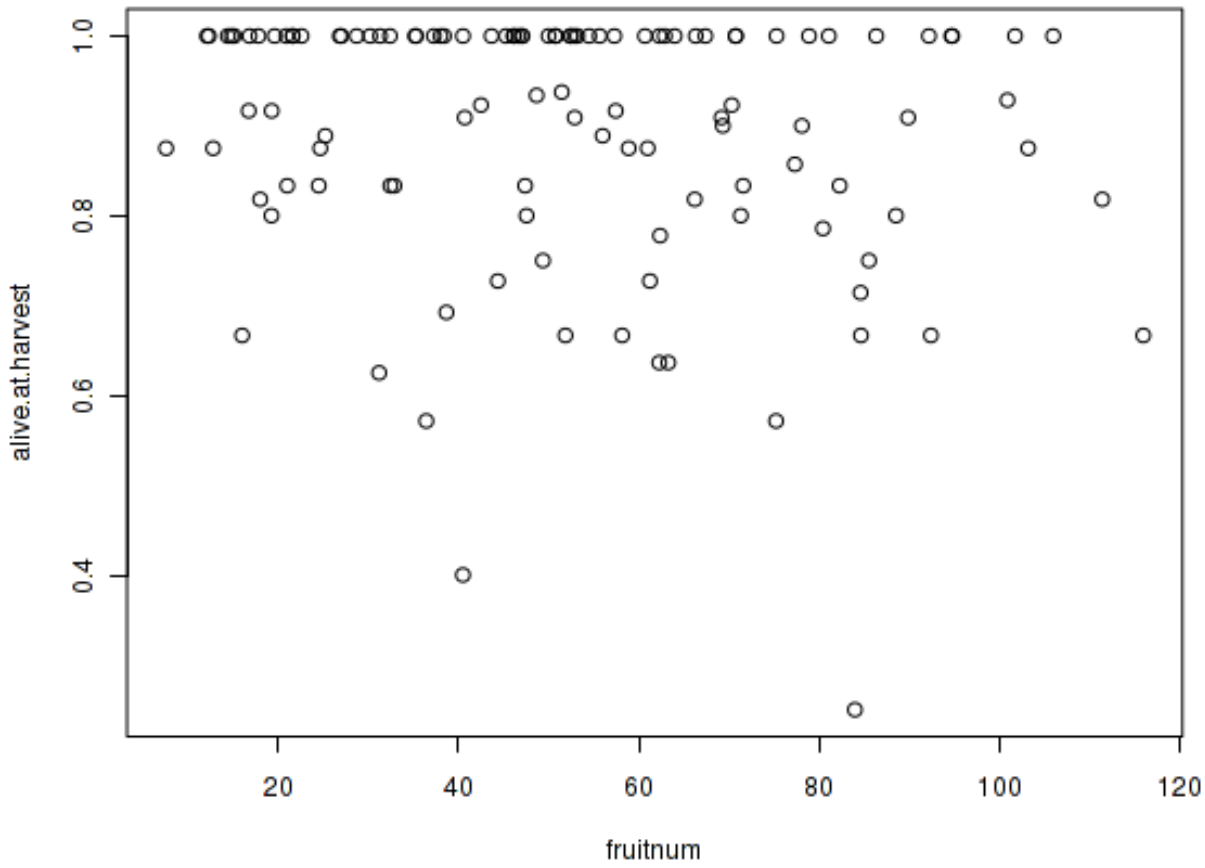
1.2.1 Some diagnostics

Does variance within a line depend on fruitnumber? This is the classic variance/mean correlation

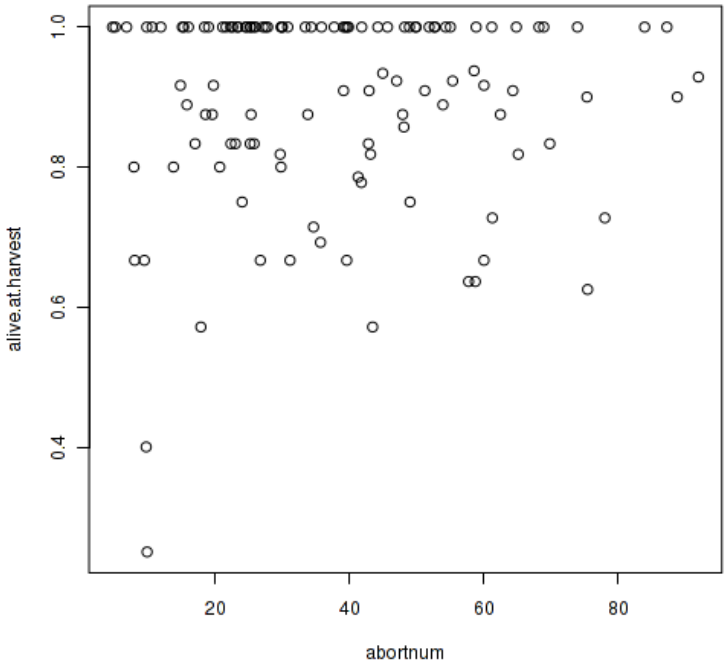
The answer in the figure below is that variance does depend on mean.



is there a reason to expect that survival rate in a line is related to average fruitnum for the line? Looks like if there is any pattern, it is the low fruitnumbers that have the high survival.

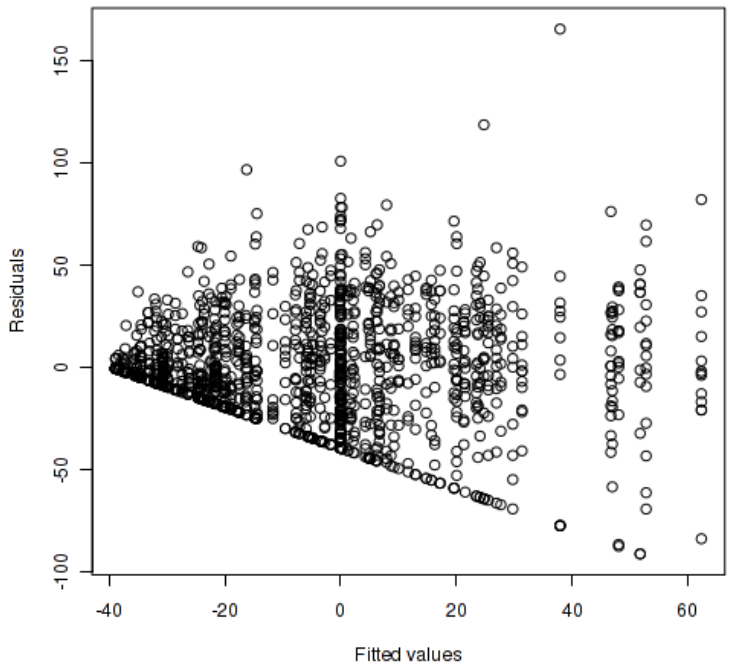


Does look like there is a similar pattern in abort/alive-at-harvest relationship

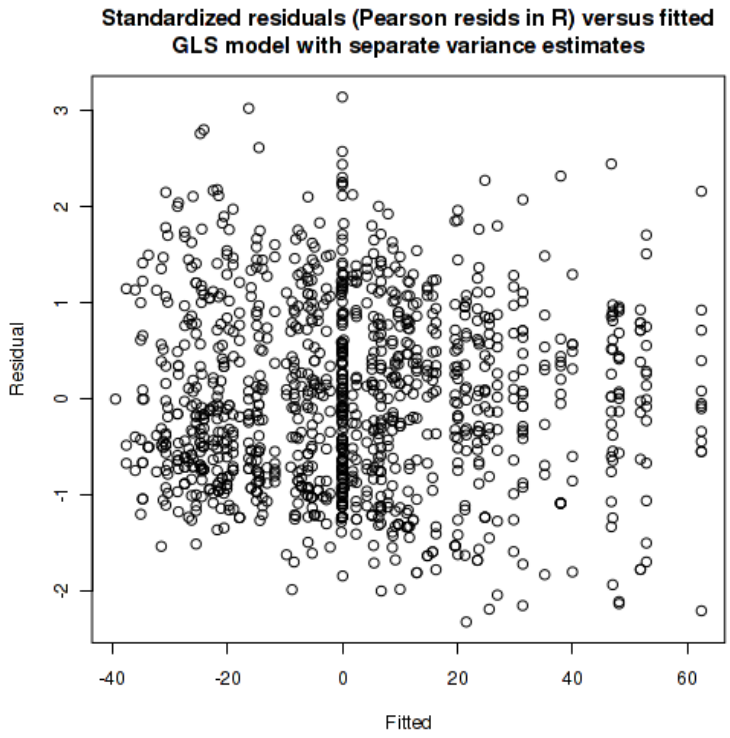


1.2.2 Crafting the best linear model to underly a testing framework

There is going to be heteroschedacity as established above. The figure below shows the residual by fitted response for a plain OLS anova: fitness as a function of line. Clearly there is terrible heteroscedacity



This figure shows the same plot using standardized residuals after using a weighting scheme that allows each line to have a separate variance (this does add tons of parameters to the model, however). You can see that the heteroschedacity disappears.



Is the addition of all the extra parameters worth it? They dramatically improve residual performance. The following likelihood ratio tests suggest that the extra parameters still produces a much better model. "fit.gls" is a "standard anova", "fit.gls.id" is the same model with individual line variance weighting.

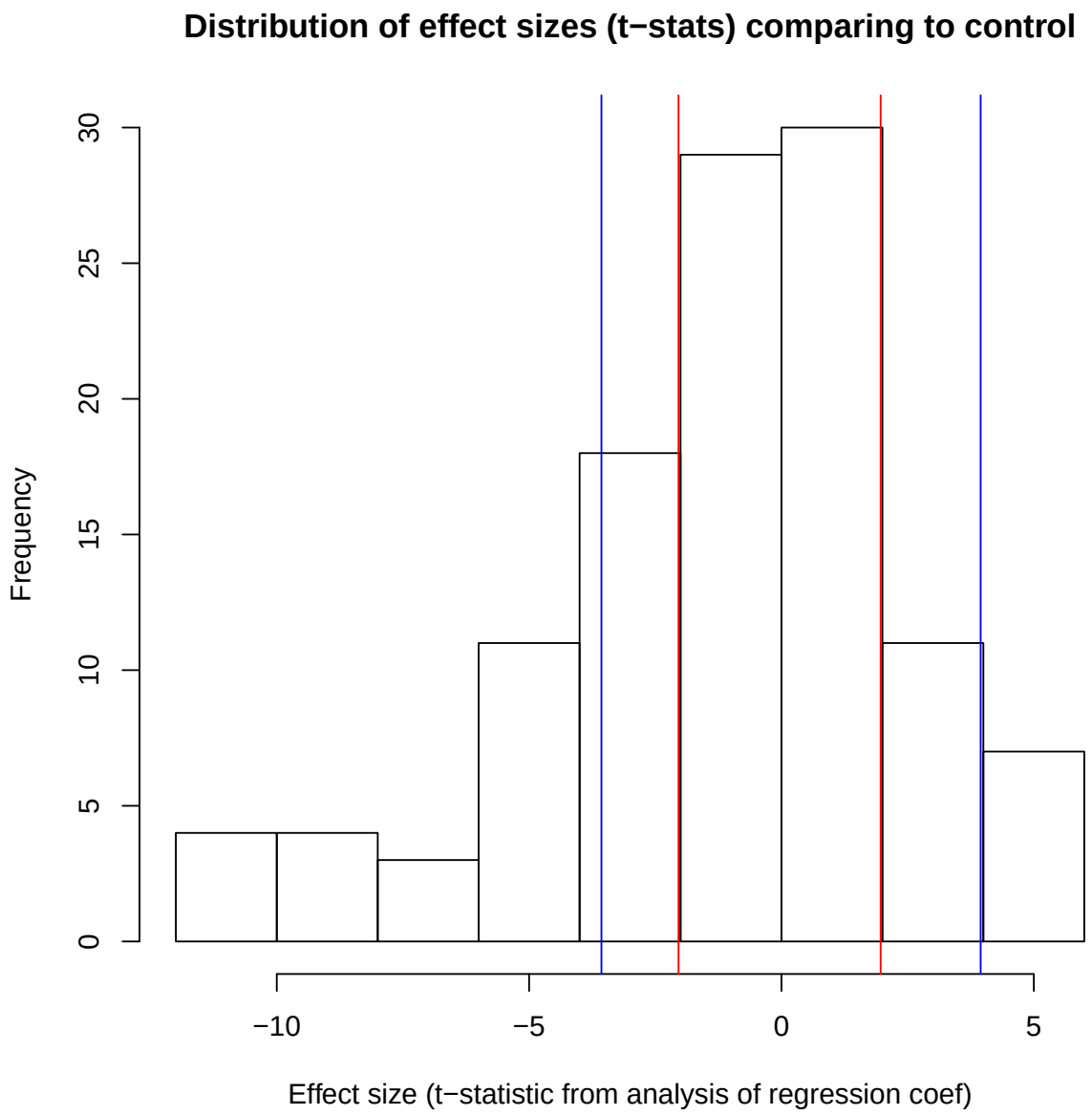
```
##            Model  df      AIC      BIC    logLik   Test  L.Ratio p-value
## fit.gls        1 118 10557.34 11141.99 -5160.671
## fit.gls.id     2 234 10392.93 11552.31 -4962.464 1 vs 2 396.4131  <.0001
## Denom. DF: 1048
##             numDF  F-value p-value
## (Intercept)     1 3336.923  <.0001
## SALK_Line     116   14.631  <.0001
```

1.2.3 Multiple test approach using treatment contrasts

First, we could look at dummy contrast coding comparing each line to the control. Technically this could be thought of as a planned comparison. Hard for me to swallow this and I expect the same is true for reviewers.

I've produced a histogram of effect sizes calculated as t-statistics in comparison to the control. The interval between the blue lines indicates lines that are not different than controls after correcting the significance of these t-stats using Bonferroni. There are 28 lines to the left of the blue interval and 8 to the right of the blue interval. The interval among the red

line corresponds to no correction for multiple tests. In this case, 40 lines are lower than the red interval and 19 are larger.



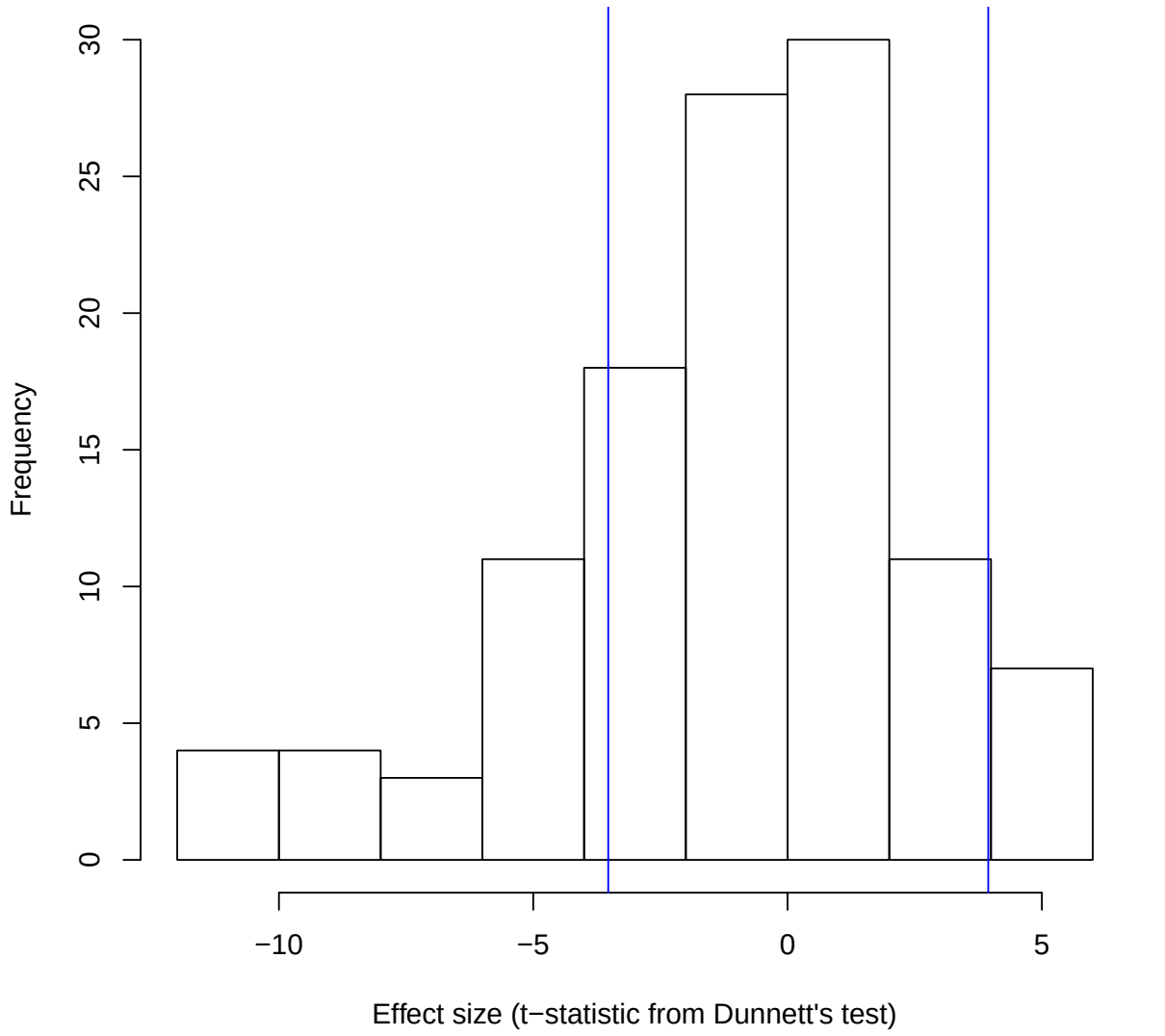

```
## [1] "Numbers of lines that have more extreme fitness than control: bonferroni correct
##
## bonsig  FALSE TRUE
##   FALSE    40   41
##   TRUE     28    8
## [1] "Numbers of lines that have more extreme fitness than control: no correction"
##
```

```
## sig     FALSE TRUE
##   FALSE    28   30
##   TRUE     40   19
```

1.2.4 Multiple test approach using Dunnett's many to one test

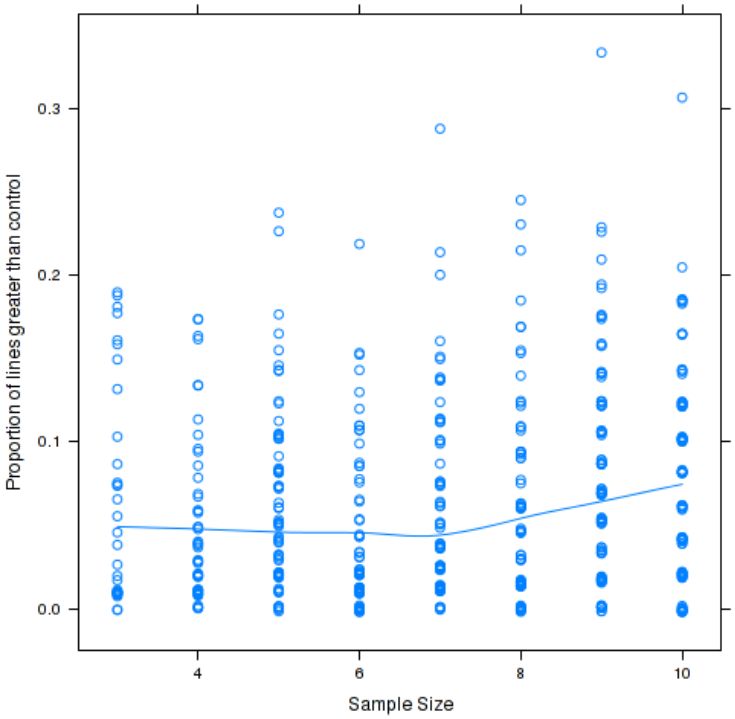
The classic post-hoc test for comparing many treatments to a control is Dunnett's test. It corrects for multiple tests. I'm not totally sure that the dunnett test is appropriate with variance weighting, but I've seen other people do it. Ultimately, I might be more confident using the straight bonferroni approach from above. Anyway, the results are not really different using this model: 28 less than control, 8 greater than control.

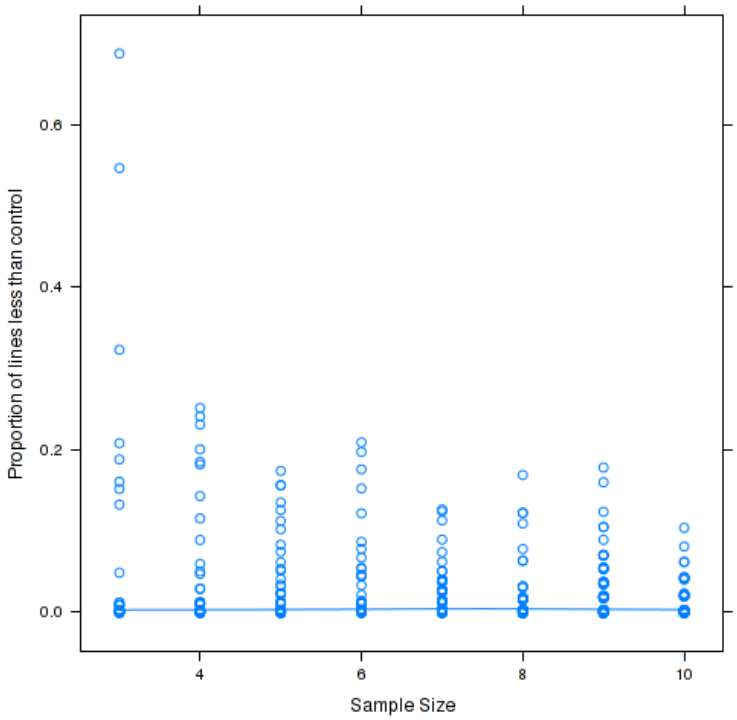
```
## Warning in RET$pfunction("adjusted", ...):  Completion with error > abseps
## Warning in RET$pfunction("adjusted", ...):  Completion with error > abseps
## Warning in RET$pfunction("adjusted", ...):  Completion with error > abseps
```




```
## [1] "Numbers of lines that have more extreme fitness than control: dunnett's"
##
## sig     FALSE TRUE
##   FALSE    39   40
##   TRUE     29    8
```

1.2.5 "power analysis" with contrasts

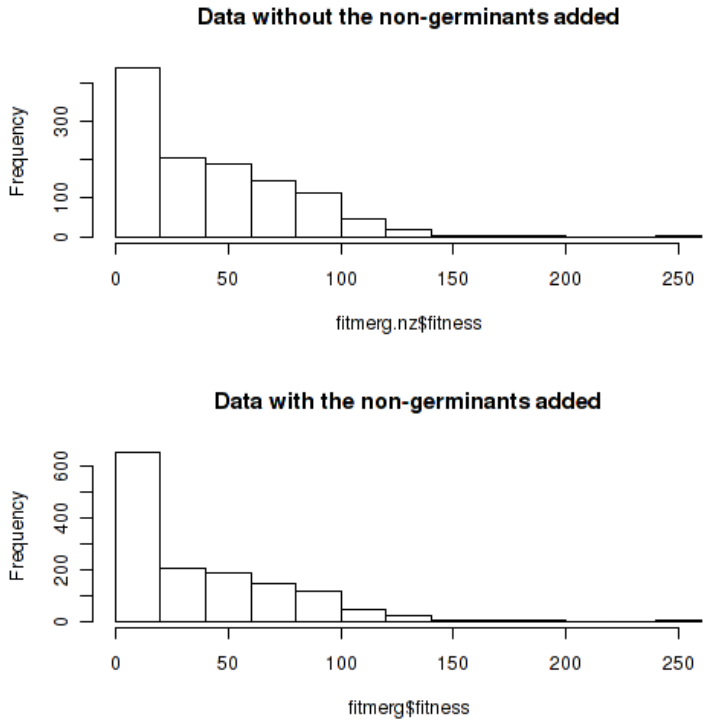

Proportion of lines greater than control
0.0
0.1
0.2
0.3
4
6
8
10
Sample Size


Proportion of lines less than control
0.0
0.2
0.4
0.6
4
6
8
10
Sample Size

1.3 Zero inflated models

These data do include the non-germinated seed categories. It is definitely these zeros that cause all the residual behavior described above.

Here are the data with the non-germinants added in and not added in:

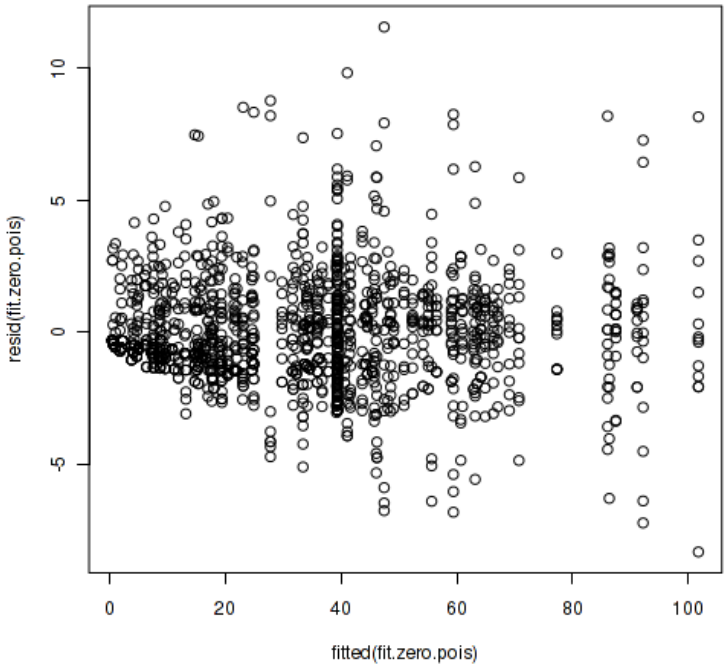


So rather than relying on the ridiculously over-parameterized independent variance models above, I decided to bite the bullet and start working with zero-inflated models.

One could argue convincingly that many of the zero fitnesses in these data result from more than one process (failure to germinate versus failure to survive and reproduce, for example). In situations like this a mixture model might be the way to go. Two such models (zero inflated poisson and zero inflated neg binomial) are fit to these data and the ad-hoc test below indicates that the negative binomial works better.

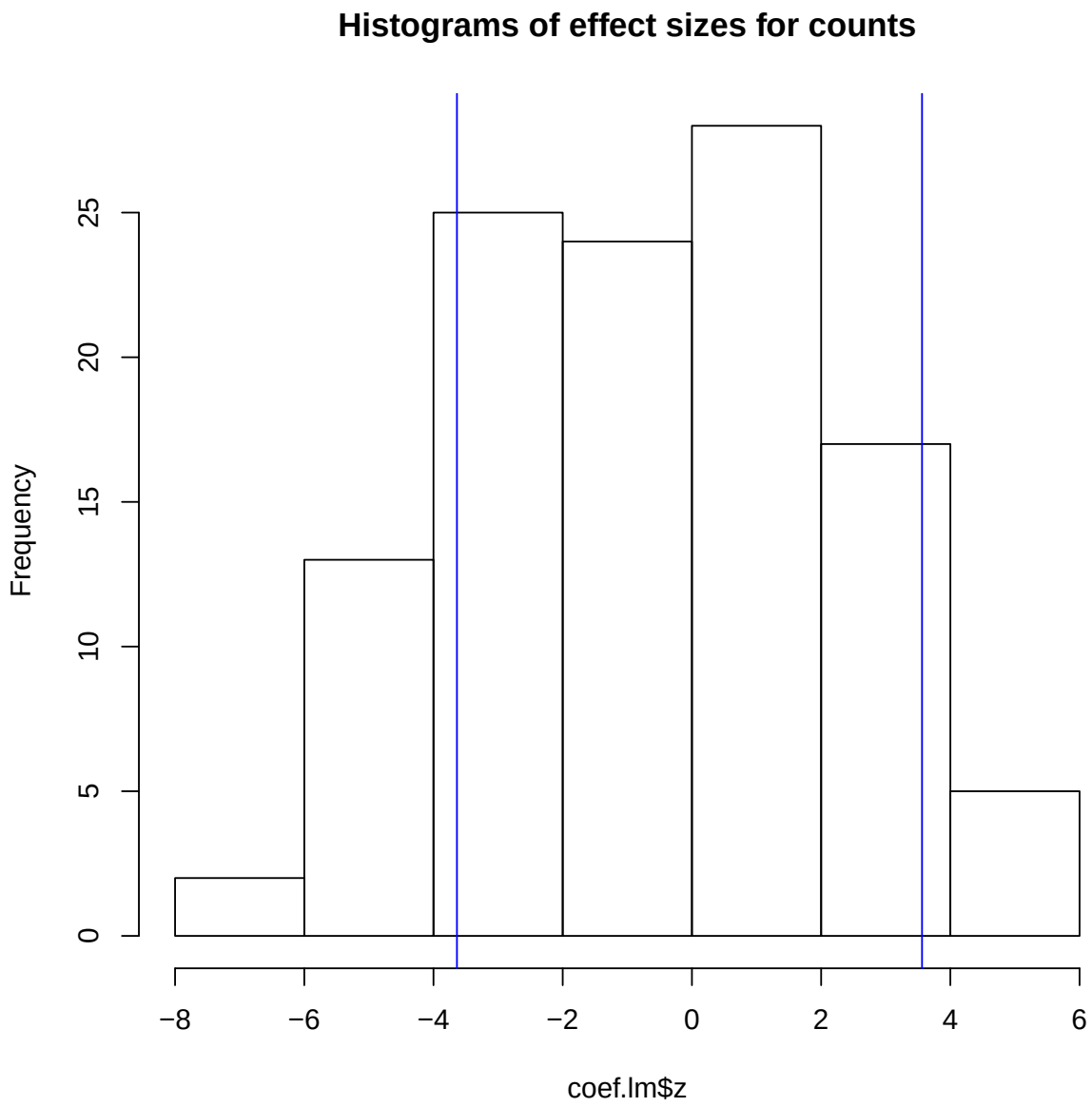
```
## Vuong Non-Nested Hypothesis Test-Statistic:
## (test-statistic is asymptotically distributed N(0,1) under the
##  null that the models are indistinguishible)
## -------------------------------------------------------------
##                Vuong z-statistic             H_A    p-value
## Raw                   -16.52004 model2 > model1 < 2.22e-16
## AIC-corrected         -16.52004 model2 > model1 < 2.22e-16
## BIC-corrected         -16.52004 model2 > model1 < 2.22e-16
```

Check out the residual/fitted plot for the fitness count portion

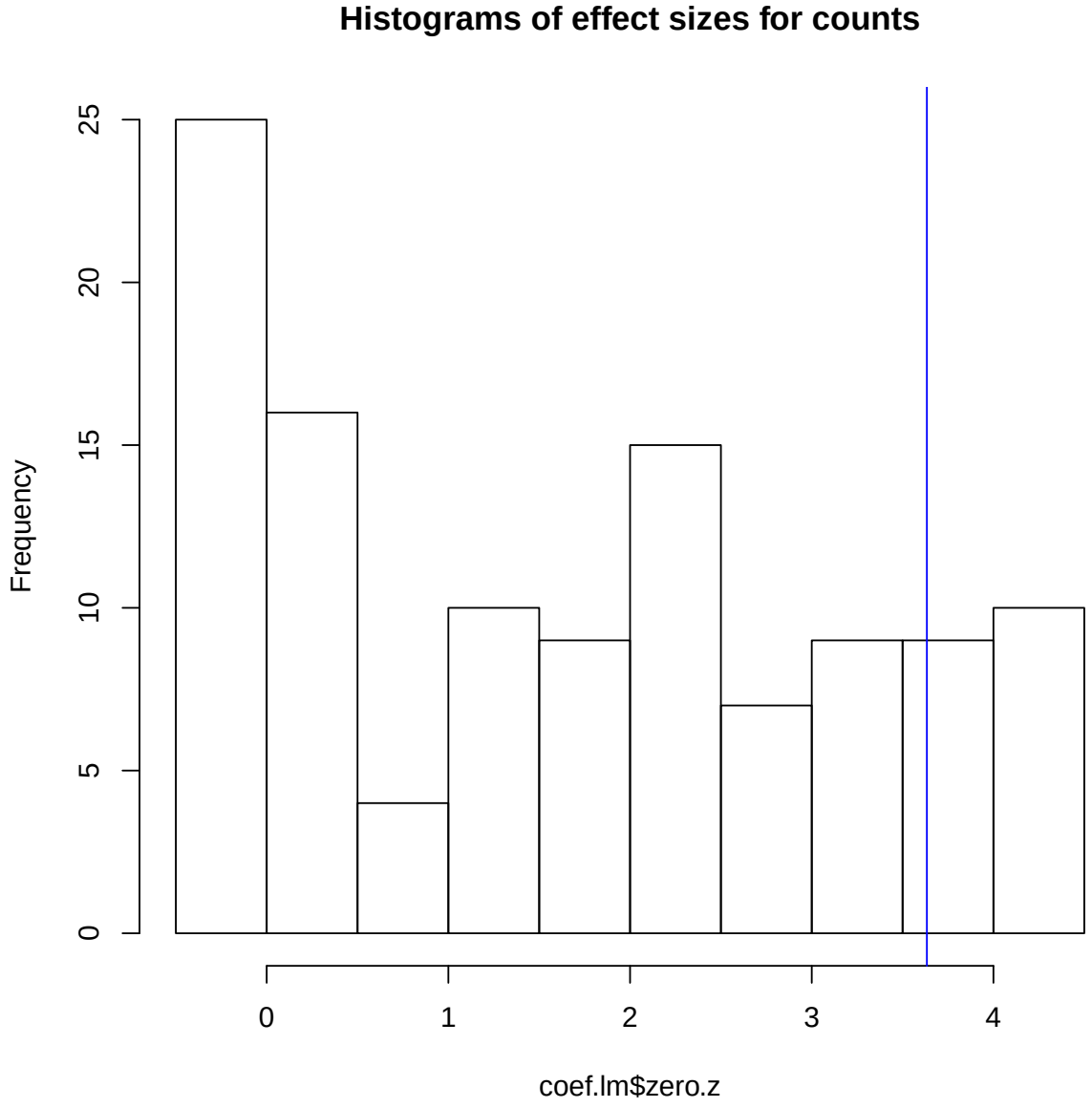


So we can perform a means-comparisons analysis like the ones described above for the zero-inflated data. Things are a little more complicated for a couple of reasons: 1) there are actually two models fit, a count model and an inflation model. We could interpret them as the effect of line on the number of fruit after germination and the effect of line on germination. This latter model is a little troubling though, because the germination rates are correlated due to common germination pot. I don't put much stock in it. 2) there are not really post-hoc tests developed for zero-inflated models so the bonferroni correction on contrasts is the only real game in town.

When using the bonferroni correction method, there are 20 lines below (17%) and 7 lines greater (6%). So lower than the mean-focused approach, but still nearly 1/4 show "phenotypes".




```
## above below
##     7    20
```

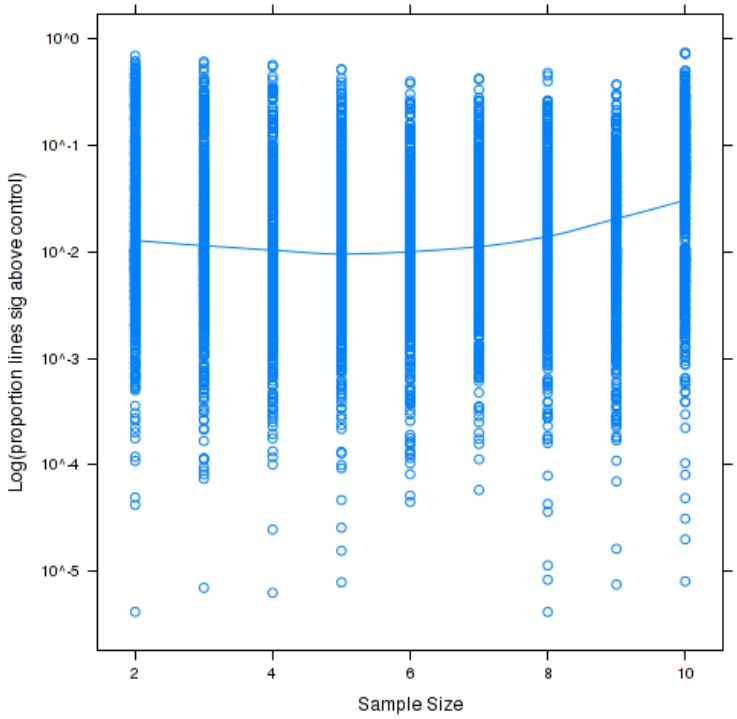


1.4 Posthoc power analysis–zero inflated model

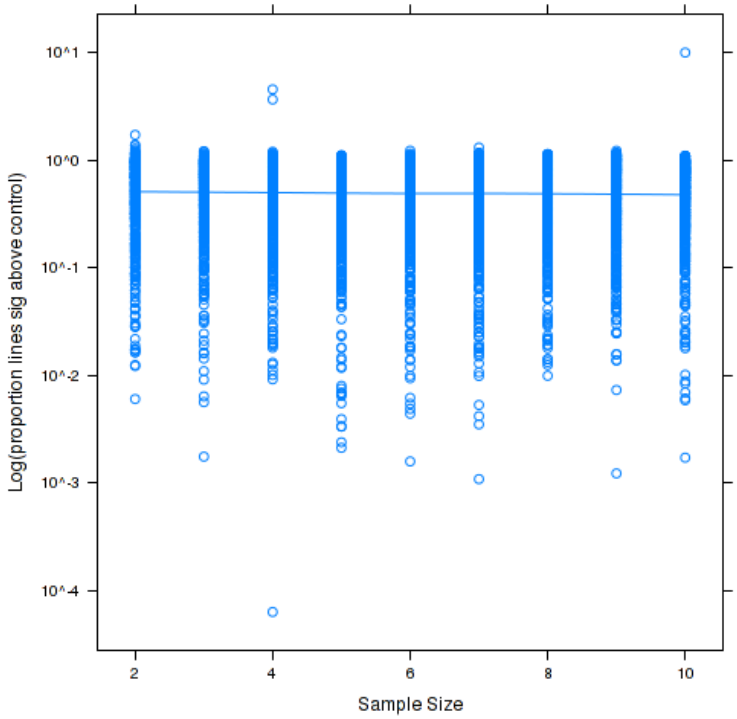
What follows is basically the same power analysis performed above on the gls models.

The following two plots show the relationship between the number of plants grown and the ability to detect the proportion of plants that have fitness above and below the control. Note the inflection at a sample size of 7 for the plants with fitness greater than controls.

```
## Warning in xyplot.formula((jitter(above, 500) + 0.001) ~ ss, type = c("p",
:  NaNs produced
```

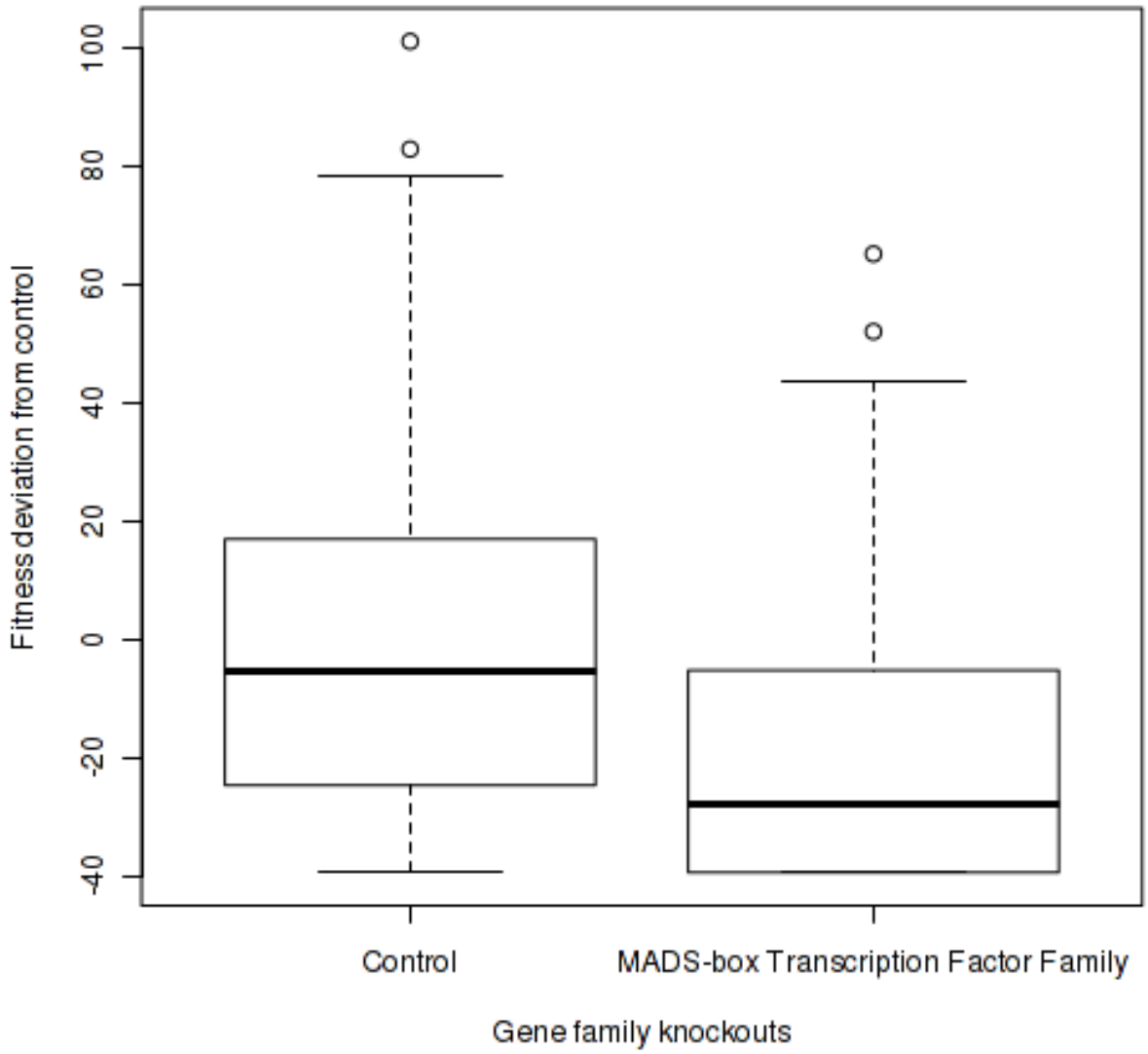



```
## Warning in xyplot.formula((jitter(below, 500) + 0.001) ~ ss, type = c("p",
:  NaNs produced
```



1.5 Specific comparisons among a-priori-chosen subsets of genes

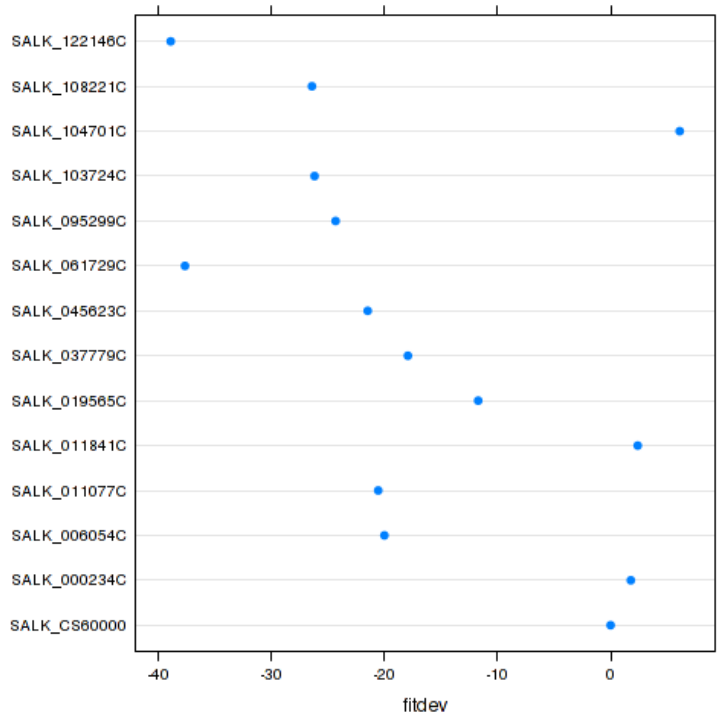
Here is a comparison of the controls to all mads-box genes




```
##                 Df Sum Sq Mean Sq F value   Pr(>F)
## GeneFamilyName   1  19866   19866   24.09 1.65e-06 ***
## Residuals      253 208645     825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Welch Two Sample t-test
##
## data:  fitdev by GeneFamilyName
```

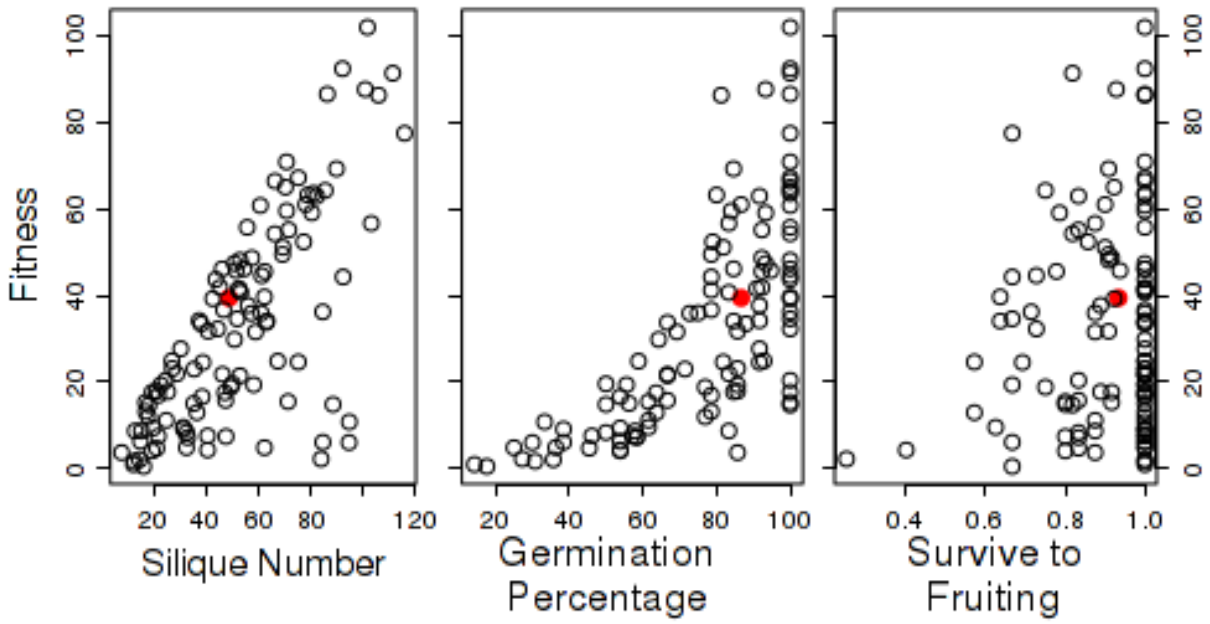
```
## t = 4.8493, df = 227.46, p-value = 2.297e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  10.49358 24.85851
## sample estimates:
##                                  mean in group Control
##                                          -1.935837e-15
## mean in group MADS-box Transcription Factor Family
##                                          -1.767604e+01
```

Here is the distribution of fitness effects among mads box genes along with a test of among-line differences




```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## SALK_Line    13  45635    3510   4.626 4.23e-07 ***
## Residuals   241 182877     759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.6 Fitness components



2 Analysis of fitness

Taking all SALK lines and pooling them and comparing to the control:

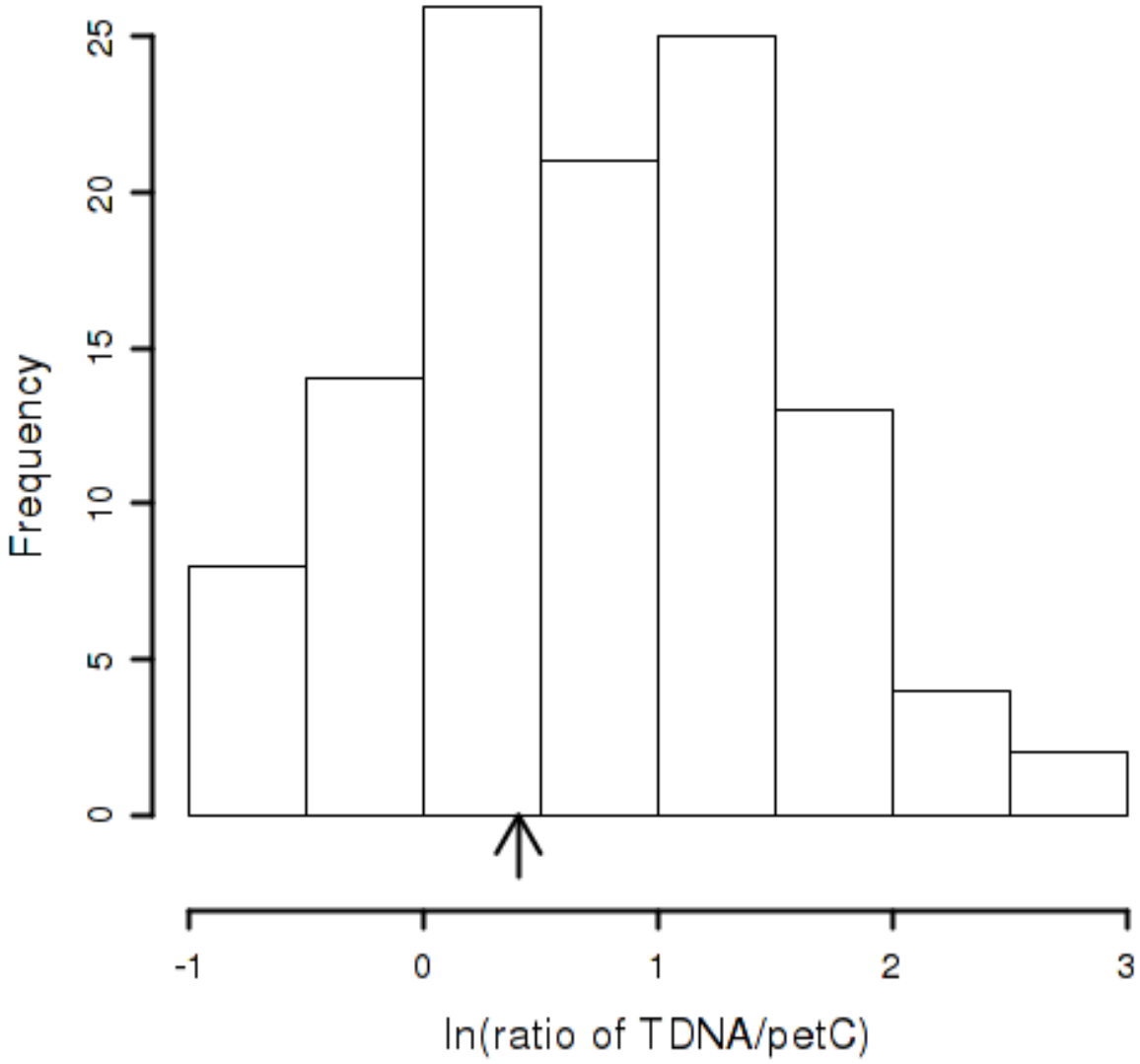
You can see from the resampled line distributions that salk lines and controls have similar mean fitnesses, with definite differences in variance among groups

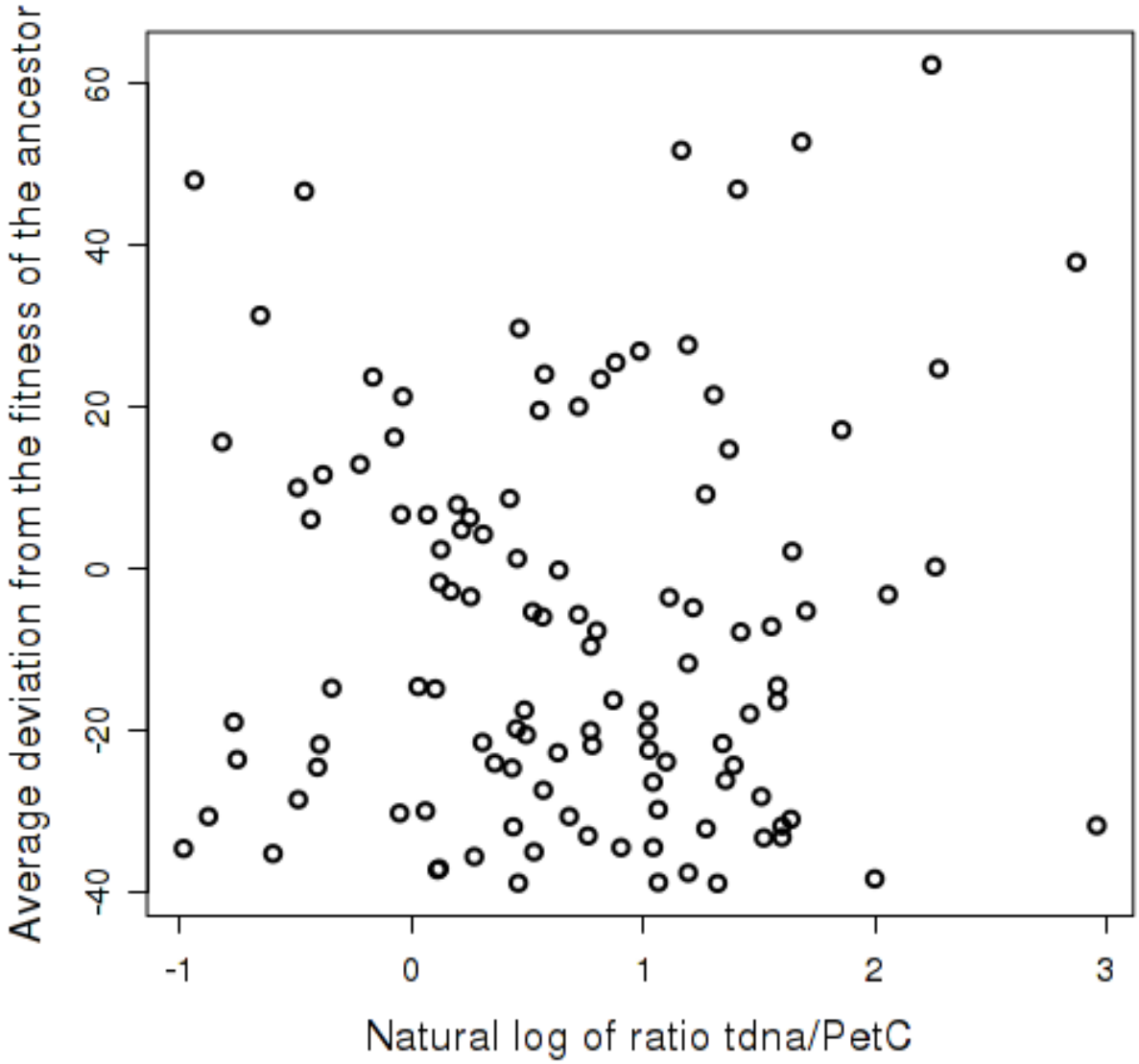
```
##
##  Welch Two Sample t-test
##
## data:  fitness by treat
## t = 1.7031, df = 150.93, p-value = 0.09061
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8436463 11.3795190
## sample estimates:
## mean in group control   mean in group treat
##              39.39545              34.12752
##
##  Kruskal-Wallis rank sum test
##
## data:  fitmerg$treat and fitmerg$fitness
## Kruskal-Wallis chi-squared = 1230.6, df = 737, p-value < 2.2e-16
```

```
## [1] "permutation typeI prob:"
## [1] 0.029
```

3 Effects of tdna insert number

Here is a plot that relates our total measure of fitness in the lines used in the original pilot study to the ratio of tdna to endogenous genes






```
Call:
lm(formula = fitdev ~ log(area.ratio), data = fruit.by.line)

Residuals:
   Min     1Q Median     3Q    Max
-32.25 -20.27  -7.08  15.91  68.48

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)      -7.4773     3.0317  -2.466   0.0152 *
log(area.ratio)   0.6092     2.8252   0.216   0.8297
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.61 on 111 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.0004187,Adjusted R-squared:  -0.008587
F-statistic: 0.04649 on 1 and 111 DF,  p-value: 0.8297
```

Clearly no pattern there and a regression confirms.

Just to make sure, I also lumped the ratios into categories and looked for a pattern again:

When focusing on medians, it looks a little bit like there might be some variation across categories

```
summary(aov(fitdev~as.factor(ratiocat),fruit.by.line))

                     Df Sum Sq Mean Sq F value Pr(>F)
as.factor(ratiocat)   2    238   119.1   0.195  0.823
Residuals           110  67020   609.3
3 observations deleted due to missingness
```

Alas, no pattern there either. I might suggest that, at the least, we worry about copy number less than other factors when choosing lines for UnPAK projects

4 Fitness as a function of gene family size

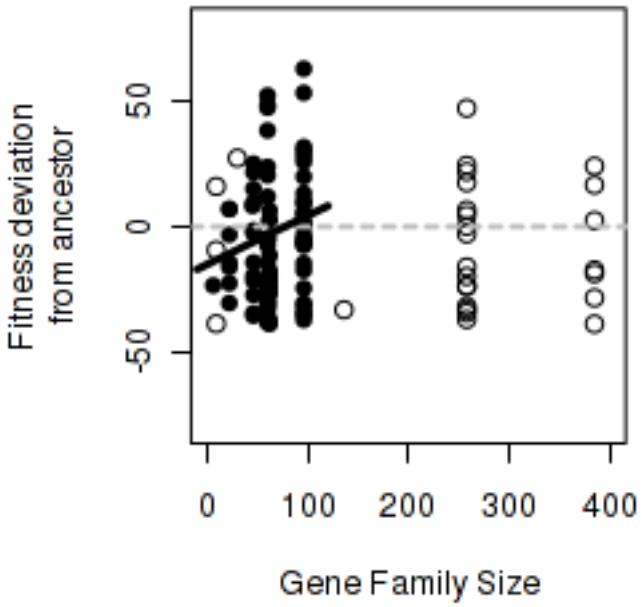
In the original pilot study we chose genes from different families with differing sizes. Again, there does not seem to be a significant relationship between gene family size and reproductive output in the non-regulatory genes, but there does seem to be a slight pattern in the regulatory genes.

This figure is based on the number of genes in the Gene Family data on tair

This is the same figure but with the means of each line plotted instead of all the reps for each line.

```
Call:
lm(formula = fitdev ~ FamilySize * Regulatory, data = fruit.by.line)

Residuals:
    Min      1Q  Median      3Q     Max
-36.460 -18.848  -4.868  16.228  63.166
```

```
Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               -5.051612  11.090659  -0.455    0.650
FamilySize                -0.009073   0.040285  -0.225    0.822
Regulatoryyes            -14.535775  13.532380  -1.074    0.285
FamilySize:Regulatoryyes   0.204854   0.118243   1.732    0.086 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.31 on 111 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.02772,Adjusted R-squared:  0.001446
F-statistic: 1.055 on 3 and 111 DF,  p-value: 0.3713

Call:
lm(formula = fitdev ~ FamilySize, data = fruit.by.line, subset = Regulatory ==
    "yes")

Residuals:
    Min      1Q  Median      3Q     Max
-36.460 -17.802  -4.868  13.646  63.166

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.5874     7.6991  -2.544   0.0128 *
FamilySize    0.1958     0.1104   1.774   0.0797 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.14 on 85 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.03569,Adjusted R-squared:  0.02434
F-statistic: 3.146 on 1 and 85 DF,  p-value: 0.0797
```



And some analyses on the means of each line's fitness deviation from the ancestor: First OLS ancova. Then regressions for non-regulatory and then regulatory genes

```
summary(aov(fitdev~FamilySize*Regulatory,fruit.by.line))

                      Df Sum Sq Mean Sq F value Pr(>F)
FamilySize             1     12    12.2   0.021  0.886
Regulatory             1     85    84.5   0.143  0.706
FamilySize:Regulatory  1   1774  1774.2   3.002  0.086 .
Residuals            111  65611   591.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
1 observation deleted due to missingness

summary(lm(fitdev~FamilySize,subset=Regulatory=="no",fruit.by.line))


Call:
lm(formula = fitdev ~ FamilySize, data = fruit.by.line, subset = Regulatory ==
    "no")

Residuals:
    Min      1Q  Median      3Q     Max
-33.697 -21.148  -6.647  21.757  54.122
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.051612  11.342875  -0.445    0.660
FamilySize  -0.009073   0.041201  -0.220    0.827

Residual standard error: 24.87 on 26 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.001862,Adjusted R-squared:  -0.03653
F-statistic: 0.04849 on 1 and 26 DF,  p-value: 0.8274
```

```
summary(lm(fitdev~FamilySize,subset=Regulatory=="yes",fruit.by.line))
```

```

Call:
lm(formula = fitdev ~ FamilySize, data = fruit.by.line, subset = Regulatory ==
    "yes")

Residuals:
    Min      1Q  Median      3Q     Max
-36.460 -17.802  -4.868  13.646  63.166

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.5874     7.6991  -2.544   0.0128 *
FamilySize    0.1958     0.1104   1.774   0.0797 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 24.14 on 85 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.03569,Adjusted R-squared:  0.02434
F-statistic: 3.146 on 1 and 85 DF,  p-value: 0.0797
```

```
summary(lm(log(fitdev+40)~FamilySize,subset=Regulatory=="yes",fruit.by.line))
```

```

Call:
lm(formula = log(fitdev + 40) ~ FamilySize, data = fruit.by.line,
    subset = Regulatory == "yes")

Residuals:
    Min      1Q  Median      3Q     Max
-3.0410 -0.5243  0.2092  0.6873  1.3955
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.843184   0.308385   9.220 1.94e-14 ***
FamilySize  0.004679   0.004421   1.058    0.293
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9669 on 85 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.013,Adjusted R-squared:  0.001393
F-statistic:  1.12 on 1 and 85 DF,  p-value: 0.2929
```

Here is a little more sophisticated analysis of the family size effect

```
fm1 <- lme(fixed=fitdev~FamilySize,random=~1|SALK_Line,method="ML",subset=Regulatory=="y
#plot(fm1)
vfix <- varFixed(~FamilySize)
fm2 <- lme(fixed=fitdev~FamilySize,random=~1|SALK_Line,method="ML",subset=Regulatory=="y
#plot(resid(fm2,type="pearson")~fitted(fm2))
fm3 <- lme(fixed=fitdev~1,random=~1|SALK_Line,method="ML",subset=Regulatory=="yes",na.ac

anova(fm3,fm1,fm2) #looks like fm1 is the best model

##     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fm3     1  3 9234.484 9249.057 -4614.242
## fm1     2  4 9233.284 9252.714 -4612.642 1 vs 2 3.200534  0.0736
## fm2     3  4 9309.930 9329.360 -4650.965

anova(fm1)

##             numDF denDF  F-value p-value
## (Intercept)     1   864 6.421385  0.0115
## FamilySize      1    85 3.255464  0.0747

anova(fm2)

##             numDF denDF  F-value p-value
## (Intercept)     1   864 6.947735  0.0085
## FamilySize      1    85 3.250567  0.0749

anova(fm3)

##             numDF denDF  F-value p-value
## (Intercept)     1   864 6.207429  0.0129
```

Ok, here is the analysis of family size with line as random intercept. Terrible heteroscedacity, repaired using a fixed variance structure. No signal of family size in the final model.

Now, the joint categories approach: Two tests. The first assumes that the rows and columns are independent, but the expected values come from the marginal totals. The second assumes that the number of fitneses in each of the four categories is equal.

```
Pearson's Chi-squared test with Yates' continuity correction

data:  tbl
X-squared = 1.295, df = 1, p-value = 0.2551
[1] 0
```

Here is a test of the change in variance through time:

```
brks=c(0,seq(10,150,10),166)
family.size.cat <- cut(fitmerg$FamilySize,breaks=brks)
sds <- with(fitmerg,tapply(fitdev,family.size.cat,sd))
brkmid <- (brks+(c(brks[-1],166)-brks)/2)[-length(sds)]
bartlett.test(fitmerg$fitdev,family.size.cat)

##
##  Bartlett test of homogeneity of variances
##
## data:  fitmerg$fitdev and family.size.cat
## Bartlett's K-squared = 53.527, df = 6, p-value = 9.189e-10

plot(sds~brkmid)
summary(lm(sds~brkmid))

##
## Call:
## lm(formula = sds ~ brkmid)
##
## Residuals:
##    (0,10]   (20,30]   (40,50]   (50,60]   (60,70]  (90,100] (130,140]
##    -5.364    -3.338     2.014     9.348    -4.720    11.645    -9.585
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.94240    5.96626   6.192   0.0016 **
## brkmid      -0.10713    0.08187  -1.309   0.2476
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.73 on 5 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.2551,Adjusted R-squared:  0.1061
## F-statistic: 1.712 on 1 and 5 DF,  p-value: 0.2476
```

So not much change in variance, though not a lot of power either.

```
Welch Two Sample t-test

data:  fitdev by Regulatory
t = -0.56527, df = 538.64, p-value = 0.5721
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -5.889892  3.257599
sample estimates:
 mean in group no mean in group yes
       -6.262103         -4.945956
```

5 Multiple environment experiment

5.1 Figures for the 1st multi-environment experiment

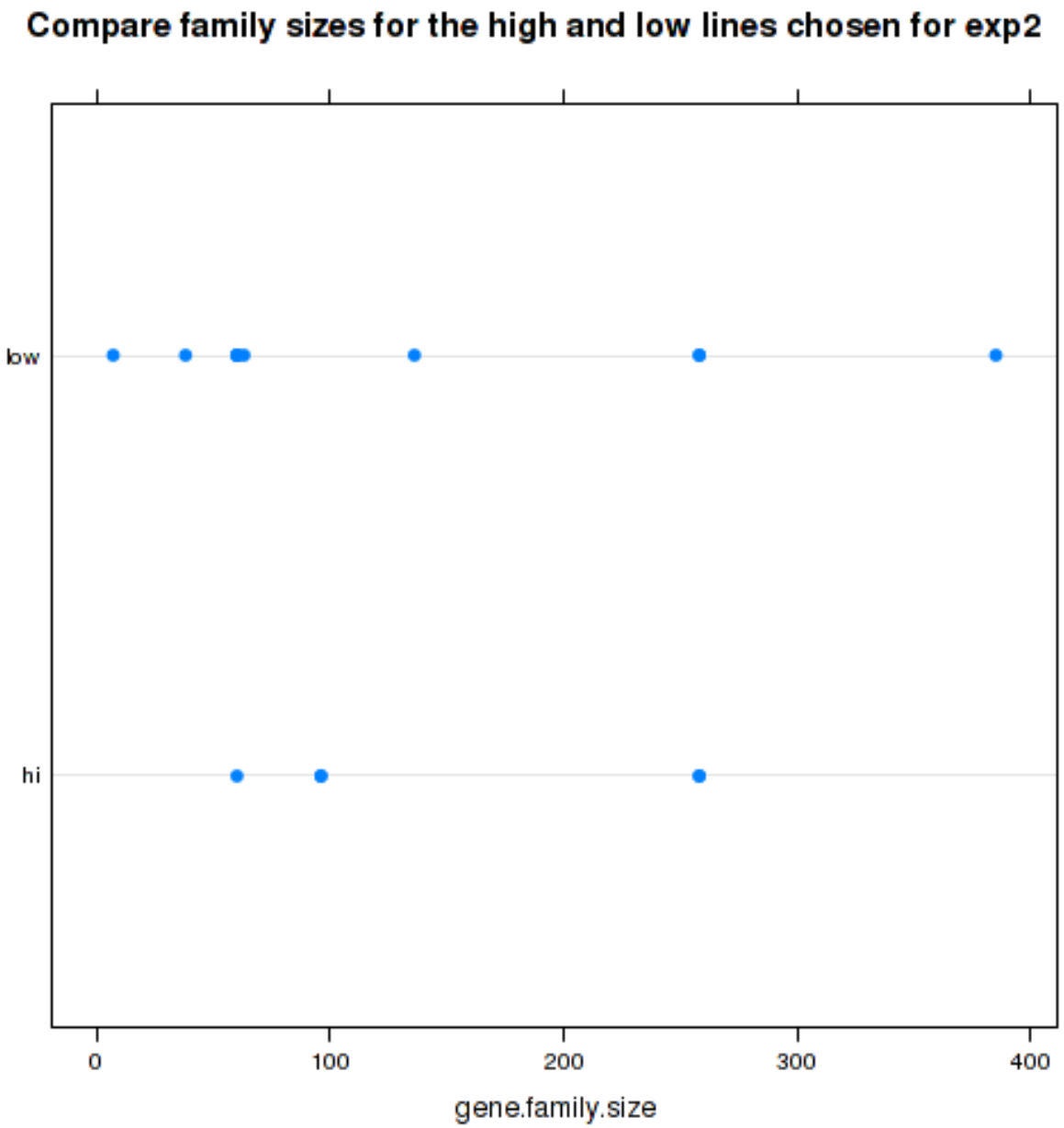
This figure is fitness deviation ignoring germination (we don't have per-sowed seed estimates (in other words, replicated) of germination for the first experiment, just average germination for that line for that germination effort. We do have those data for the second three treatments/time points

```
[1] "SALK_017933C"
[1] "SALK_033462C"
[1] "SALK_038957C"
[1] "SALK_042704C"
[1] "SALK_050488C"
[1] "SALK_054680C"
[1] "SALK_059835C"
[1] "SALK_063722C"
[1] "SALK_094332C"
[1] "SALK_126600C"
[1] "SALK_134535C"
[1] "SALK_150522C"
```

```
[1] "CS60000"
```

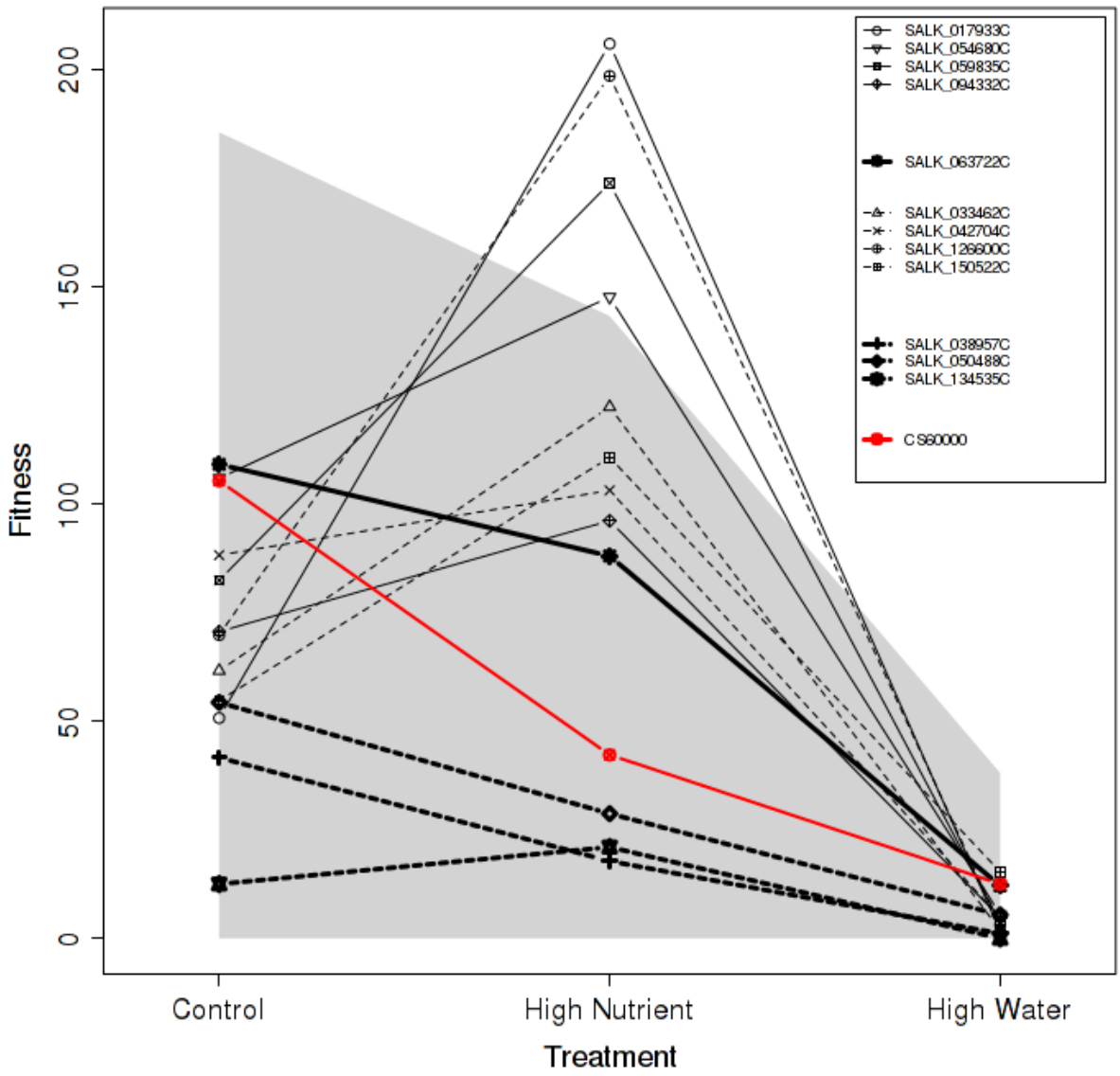
I'm going to try and address courtney's question about the gene family size of high and low lines

```
## [1] "SALK_Line"        "fitdev"           "gene.family.size"
## [4] "lohi"
```



Now here is the figure with straight mean fruit number per line:

```
[1] "SALK_017933C"
[1] 1
[1] "SALK_033462C"
[1] 1
[1] "SALK_038957C"
[1] 1
[1] "SALK_042704C"
[1] 1
[1] "SALK_050488C"
[1] 1
[1] "SALK_054680C"
[1] 1
[1] "SALK_059835C"
[1] 1
[1] "SALK_063722C"
[1] 1
[1] "SALK_094332C"
[1] 1
[1] "SALK_126600C"
[1] 1
[1] "SALK_134535C"
[1] 1
[1] "SALK_150522C"
[1] 1
[1] "CS60000"
[1] 2
```



5.1.1 Copy number

In the following figure line width is proportional to copy number category

5.2 Various tests of GxE

```
#MixedEffects
fit1 <- lmer(fitdevng~1+(1|SALK_Line),subset=SALK_Line!="CS60000",data=intresults)
```

```
fit2 <- lmer(fitdevng~treattype+(1|SALK_Line),subset=SALK_Line!="CS60000",data=intresult
fit3 <- lmer(fitdevng~treattype+treattype:SALK_Line+(1|SALK_Line),subset=SALK_Line!="CS6

anova(fit1,fit2,fit3)
```

refitting model(s) with ML (instead of REML)

```
Data: intresults
Subset: SALK_Line != "CS60000"
Models:
fit1: fitdevng ~ 1 + (1 | SALK_Line)
fit2: fitdevng ~ treattype + (1 | SALK_Line)
fit3: fitdevng ~ treattype + treattype:SALK_Line + (1 | SALK_Line)
     Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
fit1  3 4954.1 4966.3 -2474.0   4948.1
fit2  6 4898.7 4923.1 -2443.3   4886.7  61.388      3  2.970e-13 ***
fit3 50 4875.1 5078.3 -2387.5   4775.1 111.625     44  8.551e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#fit using OLS
fitaov1 <- aov(fitdevng~treattype*SALK_Line,subset=SALK_Line!="CS60000",data=intresults)
summary(fitaov1)
```

```
                     Df  Sum Sq Mean Sq F value   Pr(>F)
treattype             3  338455  112818  25.745 3.45e-15 ***
SALK_Line            11  236822   21529   4.913 3.91e-07 ***
treattype:SALK_Line  33  350555   10623   2.424 3.54e-05 ***
Residuals           382 1674009    4382
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
63 observations deleted due to missingness
```

```
#fit using ML
fitaov1.glm <-glm(fitdevng~treattype*SALK_Line,subset=SALK_Line!="CS60000",data=intresul
anova(fitaov1.glm,test="Chisq")
```

```
Analysis of Deviance Table

Model: gaussian, link: identity

Response: fitdevng
```

```
Terms added sequentially (first to last)


                    Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                    429    2599842
treattype            3   338455       426    2261387 < 2.2e-16 ***
SALK_Line           11   236822       415    2024564 1.160e-07 ***
treattype:SALK_Line 33   350555       382    1674009 8.886e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#
#now look in each environment and ask if there are line differences
#
for (trt in unique(intresults$treattype))
  {
    fit.line <- glm(fitdevng~1+as.factor(SALK_Line),subset=SALK_Line!="CS60000",data=int
    fit.intercept <- glm(fitdevng~1,subset=SALK_Line!="CS60000",data=intresults[intresul
    cat(rep("-",30));cat("\n")
    print (paste("Effect of including line in environment: ",trt))
    print(anova(fit.intercept,fit.line,test="Chisq"))
    cat(rep("-",30));cat("\n")
  }

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[1] "Effect of including line in environment:  nutrient"
Analysis of Deviance Table

Model 1: fitdevng ~ 1
Model 2: fitdevng ~ 1 + as.factor(SALK_Line)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        98    1417594
2        87    1125616 11   291978  0.02033 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[1] "Effect of including line in environment:  control"
Analysis of Deviance Table

Model 1: fitdevng ~ 1
Model 2: fitdevng ~ 1 + as.factor(SALK_Line)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
1       101     532531
2        90     439049 11    93482  0.05823 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[1] "Effect of including line in environment:  highwater"
Analysis of Deviance Table

Model 1: fitdevng ~ 1
Model 2: fitdevng ~ 1 + as.factor(SALK_Line)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        95      14671
2        84      11170 11   3500.2 0.005812 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[1] "Effect of including line in environment:  FIRST.EXP"
Analysis of Deviance Table

Model 1: fitdevng ~ 1
Model 2: fitdevng ~ 1 + as.factor(SALK_Line)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       132     296591
2       121      98174 11   198417 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Here is an analysis with Prior information and ecotypes included

```
results$genotype.cls = rep("control",dim(results)[1])
results$genotype.cls[results$fitdev<0] = "low"
results$genotype.cls[results$fitdev>0] = "high"
results$genotype.cls[results$SALK_Line=="CS60000"] = "control"
results$genotype.cls[grep("ECO",results$SALK_Line)] = "ecotype"

fit1 <- lm(fitness~treattype*genotype.cls,subset=treattype!="FIRST.EXP",data=results)
Anova(fit1,contrasts = list(treattype=contr.sum,genotype.cls=contr.sum),type=3)

Anova Table (Type III tests)
```

```
Response: fitness
                       Sum Sq  Df F value    Pr(>F)
(Intercept)            110881   1  49.774 7.596e-12 ***
treattype               44786   2  10.052 5.497e-05 ***
genotype.cls           490361   3  73.373 < 2.2e-16 ***
treattype:genotype.cls 330825   6  24.751 < 2.2e-16 ***
Residuals              895536 402
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit2 <- lm(fitness~treattype*genotype.cls,subset=(treattype!="FIRST.EXP")&(genotype.cls!
Anova(fit2,contrasts = list(treattype=contr.sum,genotype.cls=contr.sum),type=3)
```

```
Anova Table (Type III tests)

Response: fitness
                       Sum Sq  Df F value    Pr(>F)
(Intercept)            110881   1  54.756 9.043e-13 ***
treattype               44786   2  11.058 2.157e-05 ***
genotype.cls           490177   2 121.031 < 2.2e-16 ***
treattype:genotype.cls 328440   4  40.548 < 2.2e-16 ***
Residuals              759380 375
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This analysis just looks for GxE and main effects in the ecotypes. Not much signal Ecotype lines only

```
fitaov.genotype.cls.nomut <- aov(fitness~treattype*SALK_Line,subset=(treattype!="FIRST.E
summary(fitaov.genotype.cls.nomut)
```

```
##                      Df Sum Sq Mean Sq F value Pr(>F)
## treattype             2  20455   10228   1.714  0.234
## SALK_Line             9  23496    2611   0.438  0.883
## treattype:SALK_Line   9  58971    6552   1.098  0.446
## Residuals             9  53689    5965
```

```
fitall <- glm(fitdev~treattype*SALK_Line,subset=treattype!="FIRST.EXP",data=intresults)
(anova(fitall,test="F"))
```

```
## Analysis of Deviance Table
##
```

```
## Model: gaussian, link: identity
##
## Response: fitdev
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                                  383    3103317
## treattype            2   647029       381    2456288 59.5646 < 2.2e-16 ***
## SALK_Line           12   234750       369    2221538  3.6018 4.360e-05 ***
## treattype:SALK_Line 24   347732       345    1873806  2.6676 5.317e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#add in a classifier for early experiment performance
firstmns <- with(intresults[intresults$treattype=="FIRST.EXP",c("fitdev","SALK_Line")],
                 aggregate(cbind(first.fitdev=fitdev),by=list(SALK_Line=SALK_Line),mean,
intresults <- merge(intresults,firstmns,all.x=T)

fitlo <-  glm(fitdev~treattype*SALK_Line,subset=((treattype!="FIRST.EXP")&(first.fitdev<
(anova(fitlo,test="F"))

## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: fitdev
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                                  236    1805730
## treattype            2   278944       234    1526786 25.0251 1.731e-10 ***
## SALK_Line            7   146974       227    1379812  3.7673 0.0007137 ***
## treattype:SALK_Line 14   192704       213    1187108  2.4697 0.0029519 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fithi <-  glm(fitdev~treattype*SALK_Line,subset=((treattype!="FIRST.EXP")&(first.fitdev>
(anova(fithi,test="F"))
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: fitdev
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                                   146    1232611
## treattype            2   428976       144     803635 41.2297 1.228e-14 ***
## SALK_Line            4    18145       140     785490  0.8720   0.48267
## treattype:SALK_Line  8    98791       132     686698  2.3738   0.02021 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
intresults$lohi <- factor(ifelse(intresults$first.fitdev<=0,"low","high"))
fitlowhi <- glm(fitdev~treattype*lohi,subset=(treattype!="FIRST.EXP"),data=intresults)
(anova(fitlowhi,test="F"))
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: fitdev
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev       F    Pr(>F)
## NULL                             383    3103317
## treattype       2   647029       381    2456288 52.4749 < 2.2e-16 ***
## lohi            1    68562       380    2387726 11.1209 0.0009381 ***
## treattype:lohi  2    57305       378    2330421  4.6475 0.0101397 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(lm(fitdev~treattype*lohi,subset=(treattype!="FIRST.EXP"),data=intresults),contrast
```

```
## Anova Table (Type III tests)
##
## Response: fitdev
```

```
##                  Sum Sq  Df F value    Pr(>F)
## (Intercept)       23242   1  3.7699   0.05293 .
## treattype        428976   2 34.7905 1.353e-14 ***
## lohi              15911   1  2.5808   0.10900
## treattype:lohi    57305   2  4.6475   0.01014 *
## Residuals       2330421 378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fiteco <-  glm(fitdev~treattype*SALK_Line,data=intresults.ecotypes)
(anova(fiteco,test="F"))
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: fitdev
##
## Terms added sequentially (first to last)
##
##
##                     Df Deviance Resid. Df Resid. Dev      F Pr(>F)
## NULL                                    29     154737
## treattype            2    18581        27     136156 1.5574 0.2625
## SALK_Line            9    23496        18     112660 0.4376 0.8829
## treattype:SALK_Line  9    58971         9      53689 1.0984 0.4456
```

5.3 Tables for the MS

5.3.1 Gene families

```
famtable <- unique(fitmerg[!is.na(fitmerg$Regulatory),c("GeneFamilyName","FamilySize","R
linesfromfams <- with(unique(fitmerg[fitmerg$GeneFamilyName!="Control",c("SALK_Line","Ge
famtable <- merge(famtable,linesfromfams,all.x=T)
famtable <- famtable[order(famtable$Regulatory,-famtable$FamilySize),]
famtable$Function <- ifelse(famtable$Regulatory=="yes","Regulatory","Metabolic")
famtable <- famtable[,-which(names(famtable)=="Regulatory")]
```

```
require(xtable)
```

```
## Loading required package:  xtable
##
```

```
## Attaching package:  'xtable'
##
## The following objects are masked from 'package:Hmisc':
##
##     label, label<-
```

```
print(file="redundancy-tables-figs/tbl1-genefams.html",xtable(famtable),type="html",
      include.rownames=F)
```

5.3.2 SALK Line list

```
salktbl <- unique(fitmerg[,c("SALK_Line","Gene","Gene_Family")])
salktbl <- salktbl[complete.cases(salktbl),]
salktbl <- salktbl[order(salktbl$Gene_Family,salktbl$SALK_Line),]
print(file="redundancy-tables-figs/tbls1-salk-lines.html",xtable(salktbl),type="html",
      include.rownames=F)
```

5.4 Test for effect of line on fitness for first exp.

```
summary(aov(log(fitness+1)~SALK_Line,data=fitmerg[grep("SALK.[0-9]+C",fitmerg$SALK_Line)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## SALK_Line    114   1525  13.374   6.375 <2e-16 ***
## Residuals   1144   2400   2.098
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6 Naturally occuring variants

We took the data from Cao et al 2010 and determined how many of the lines in this experiment also showed some sort of natural variation in gene function

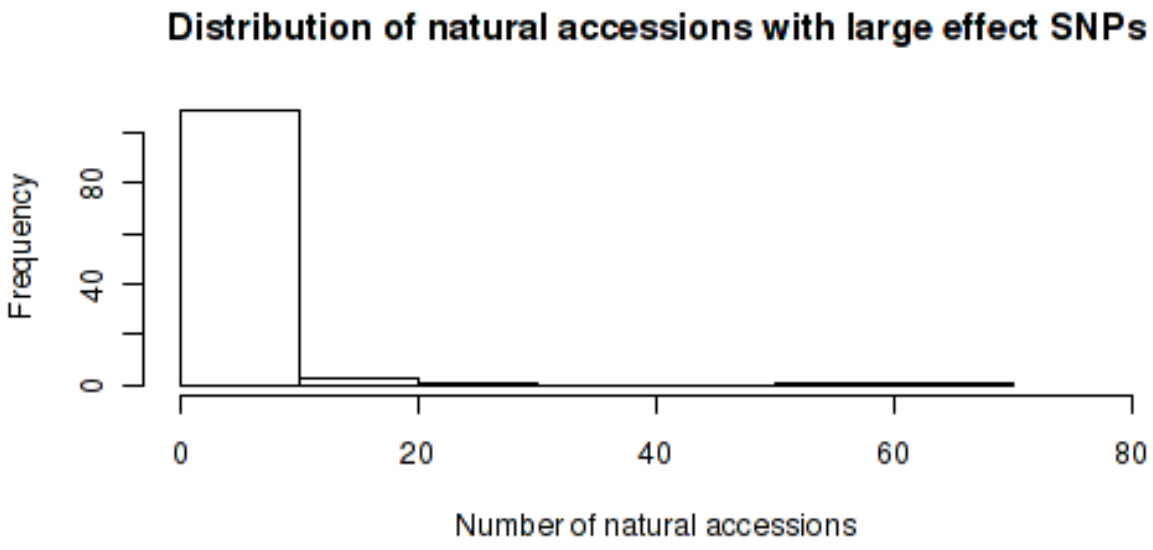
```
suppressMessages(require(dplyr))
snp <- unique(read.csv(paste0(csvdir,"/phen-snp.csv"))[,1:2])
names(snp)[2] <- "snp.strains"
snp <- snp %>% group_by(Accession) %>% summarise(snp.strains.mn=sum(snp.strains))
sv <- unique(read.csv(paste0(csvdir,"/phen-sv.csv"))[,1:2])
```

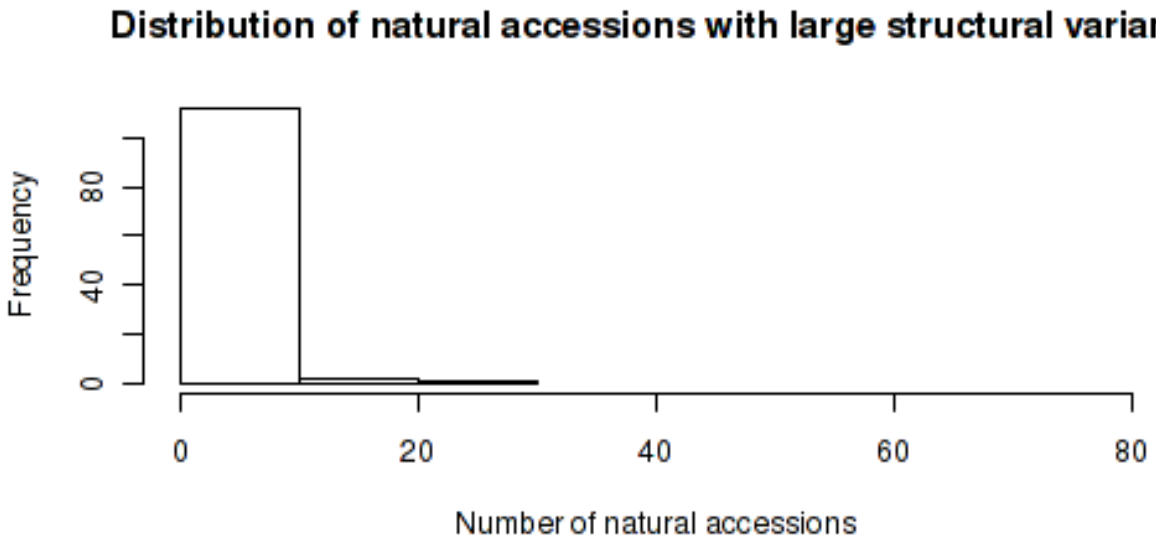
```
names(sv)[2] <- "sv.strains"
sv <- sv %>% group_by(Accession) %>% summarise(sv.strains.mn=sum(sv.strains))
write.table(file="cao-digested.csv",sep=",",row.names=F,unique(merge(snp,sv)))
```

There are definitely lines that are knocked out in nature. The first table is the frequency of lines with no natural variants (false) versus variants for SNPs that should drastically alter gene function. The second is for the distribution of lines with large structural variants

```
with(unique(snp),table(snp.strains.mn>0))

##
## FALSE  TRUE
##    99    16

with(unique(sv),table(sv.strains.mn>0))

##
## FALSE  TRUE
##   106     9
```

The following figure illustrates the distribution of the naturally occurring variants in our

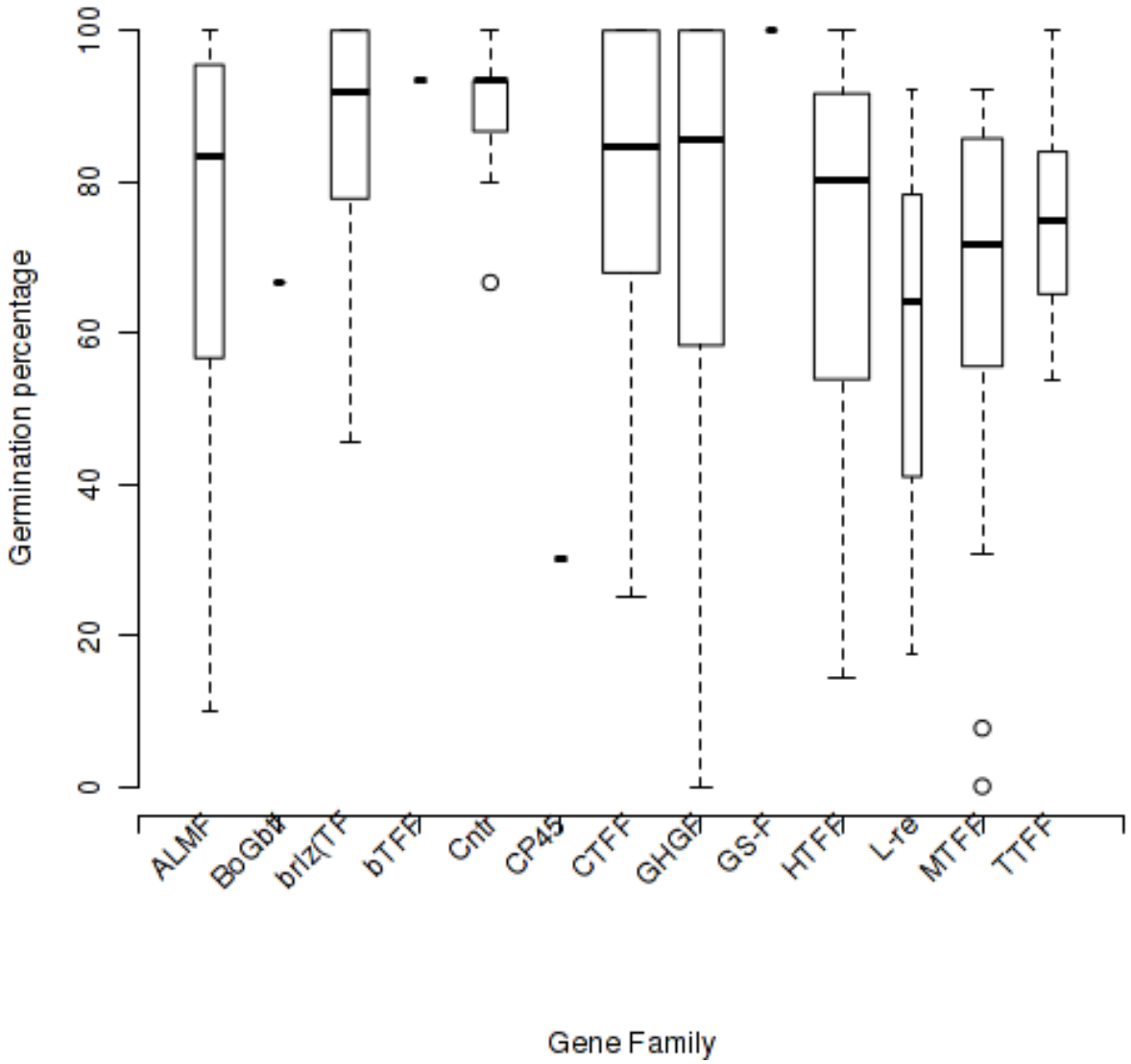




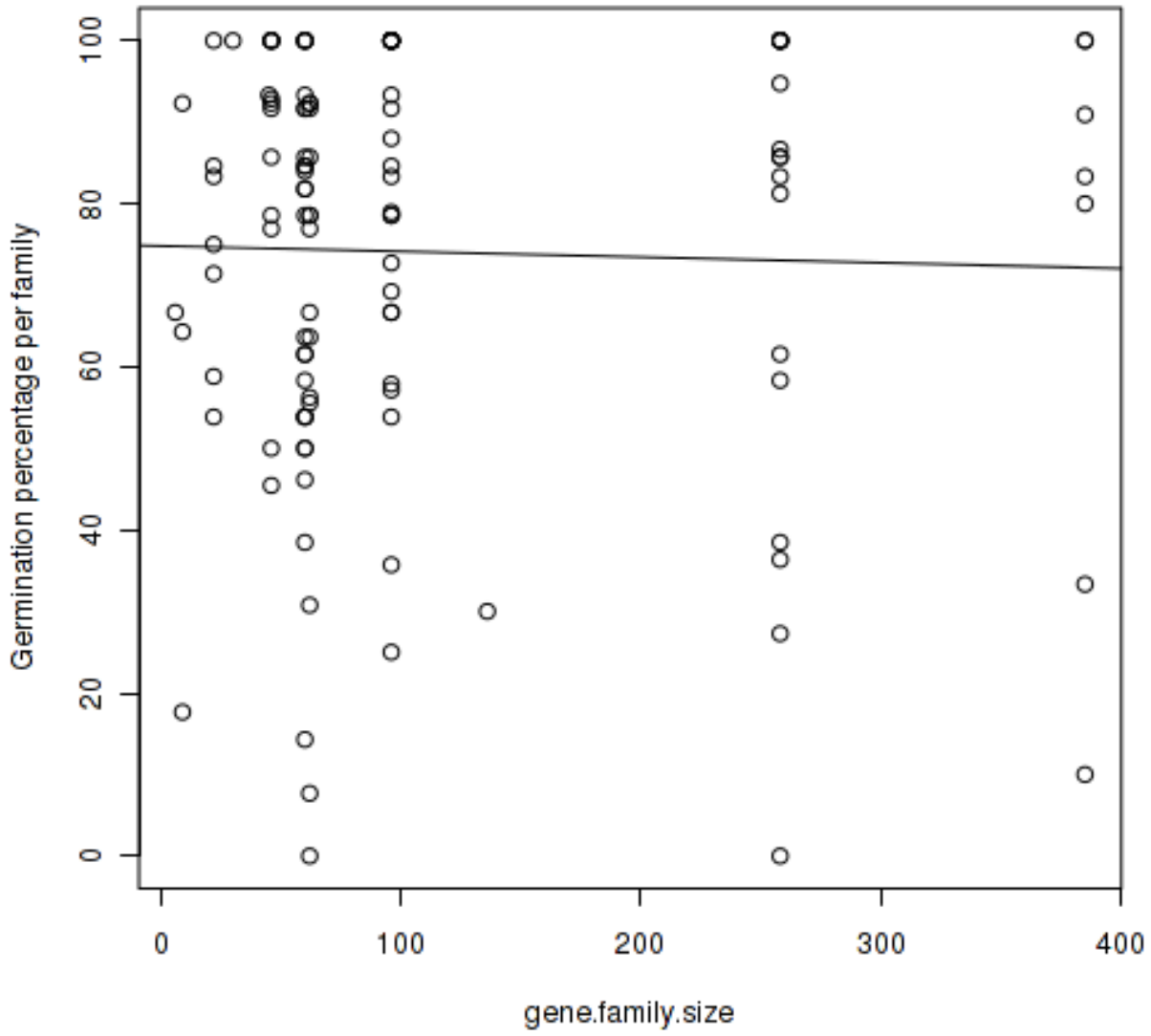
collection of lines.

6.1 Some additional comparisons we have not emphasized

6.1.1 Germination rate as a function of gene-family



Though there is variation among families in germination rate, family size does not explain it.




```
Call:
lm(formula = percent ~ (gene.family.size), data = germ.gene.families)

Residuals:
    Min      1Q  Median      3Q     Max
-74.390 -16.074   8.212  18.926  27.838

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)      74.817309   3.558032   21.03   <2e-16 ***
gene.family.size -0.006896   0.023795   -0.29    0.772
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.63 on 115 degrees of freedom
  (16 observations deleted due to missingness)
Multiple R-squared:  0.0007297,Adjusted R-squared:  -0.00796
F-statistic: 0.08398 on 1 and 115 DF,  p-value: 0.7725
```

And here's another test of the same hypothesis using permutation approach (may be less affected by unequal var).

```
[1] "typeI prob:"
[1] 0.011
```

6.2 Survival to harvest as a function of gene family

Here are analyses that examine the effect of gene family upon survival:

```
                Df Sum Sq Mean Sq F value Pr(>F)
GeneFamilyName  12 0.1829 0.01524   0.758  0.692
Residuals      105 2.1122 0.02012
```