Livestock Breeding and Genomics - Solution 4

Peter von Rohr

2022-10-14

Problem 1: QTL Data

Estimate genotypic values a and d and predict breeding values for all animals using the QTL-data given under:

/Users/pvr/Data/Projects/Github/charlotte-ngs/lbgfs2022_gh-root/main/lbgfs2022/docs/data/p1_qtl_1_loci.csv

Solution

Read the QTL data

```
tbl_qtl <- readr::read_delim(file = s_data_path, delim = ";",
                             col_types = readr::cols(
                               ID = readr::col_integer(),
                               L1_pat = readr::col_integer(),
                               L1_mat = readr::col_integer(),
                               Phenotype = readr::col_double()
                             ))
head(tbl_qtl)
```

```
## # A tibble: 6 x 4
##      ID L1_pat L1_mat Phenotype
##   <int>  <int>  <int>     <dbl>
## 1    46      2      1    -1.16
## 2    47      1      1    -0.780
## 3    48      1      2     0.345
## 4    49      2      1    -1.86
## 5    50      1      1     0.539
## 6    51      2      2    -1.16
```

Determine the QTL genotype from the maternal and the paternal allele

```
library(dplyr)
tbl_qtl_geno <- tbl_qtl %>%
  mutate(Genotype = L1_pat + L1_mat - 2L)
head(tbl_qtl_geno)
```

```
## # A tibble: 6 x 5
##      ID L1_pat L1_mat Phenotype Genotype
##   <int>  <int>  <int>     <dbl>    <int>
```

```
## 1    46      2      1    -1.16         1
## 2    47      1      1    -0.780        0
## 3    48      1      2     0.345        1
## 4    49      2      1    -1.86         1
## 5    50      1      1     0.539        0
## 6    51      2      2    -1.16         2
```

Fit the regression of phenotypes on genotypes using only the homozygous animals

```
tbl_qtl_hom <- tbl_qtl_geno %>%
  filter(Genotype != 1L)
head(tbl_qtl_hom)
```

```
## # A tibble: 6 x 5
##      ID L1_pat L1_mat Phenotype Genotype
##   <int>  <int>  <int>     <dbl>    <int>
## 1    47      1      1    -0.780        0
## 2    50      1      1     0.539        0
## 3    51      2      2    -1.16         2
## 4    52      1      1    -0.186        0
## 5    53      1      1    -1.32         0
## 6    54      1      1    -0.691        0
```

Check the genotype frequencies

```
table(tbl_qtl_hom$Genotype)
```

```
##
##   0   2
## 754 584
```

The regression is used to determine the genotypic value a

```
lm_geno_a <- lm(Phenotype ~ Genotype, data = tbl_qtl_hom)
(sry_geno_a <- summary(lm_geno_a))
```

```
##
## Call:
## lm(formula = Phenotype ~ Genotype, data = tbl_qtl_hom)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9481 -0.6912  0.0175  0.6531  3.4605
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21427    0.03684   5.817 7.51e-09 ***
## Genotype     0.01343    0.02788   0.482     0.63
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.012 on 1336 degrees of freedom
## Multiple R-squared:  0.0001737,  Adjusted R-squared:  -0.0005747
## F-statistic: 0.2321 on 1 and 1336 DF,  p-value: 0.6301
```

The genotypic value a can be read from the regression slope which is

```
n_value_a <- sry_geno_a$coefficients["Genotype", "Estimate"]
```

$$a = 0.0134306$$

The value d is obtained by subracting from the mean of the heterozygous animals, the intercept and the value a. First, we generate a dataset with only heterozygous animals.

```
tbl_qtl_het <- tbl_qtl_geno %>%
  filter(Genotype == 1L)
table(tbl_qtl_het$Genotype)
```

```
##
##    1
## 1376
```

The value d is computed as

```
n_mean_het <- mean(tbl_qtl_het$Phenotype)
n_inter <- sry_geno_a$coefficients["(Intercept)", "Estimate"]
n_val_d <-  n_mean_het - n_value_a - n_inter
```

$$d = 0.1906546 - 0.0134306 - 0.214266 = -0.0370421$$

Problem 2: Increase Effects of Genotype on Phenotype

Change the phenotypic records in the above given dataset such that the QTL explains 50 of the genetic variation when a heritability of 0.45 is assumed. It is assumed that the QTL acts purely additively, hence the genotypic value of the heterozygotes can be set to $d = 0$.

Show the results as a scatter plot of all phenotypic values for the QTL genotypes.

Solution

In a first step the phenotypic variance, the genetic variance and the QTL-variance must be determined from the data and the given information.

```
tbl_qtl <- readr::read_delim(file = s_data_path, delim = ";",
                             col_types = readr::cols(
                               ID = readr::col_integer(),
                               L1_pat = readr::col_integer(),
                               L1_mat = readr::col_integer(),
                               Phenotype = readr::col_double()
                             ))
head(tbl_qtl)
```

```
## # A tibble: 6 x 4
##      ID L1_pat L1_mat Phenotype
##   <int>  <int>  <int>     <dbl>
## 1    46      2      1    -1.16
## 2    47      1      1    -0.780
## 3    48      1      2     0.345
## 4    49      2      1    -1.86
## 5    50      1      1     0.539
## 6    51      2      2    -1.16
```

The phenotypic variance is computed as

```
(n_phen_var <- var(tbl_qtl$Phenotype))
```

```
## [1] 1.016534
```

The genetic variance is

```
(n_gen_var <- n_h2_all * n_phen_var)
```

```
## [1] 0.4574404
```

The variation explained by the QTL is

```
(n_qtl_var <- n_ratio_qtl * n_gen_var)
```

```
## [1] 0.2287202
```

From `n_qtl_var`, the additive effect a can be computed. Because, $d = 0$, the additive effect α is the same as a. Hence with $\alpha = a$, the genetic additive variance (σ^2_A) is computed as $\sigma^2_A = 2pqa^2$

From that we get

$$a = \sqrt{\frac{\sigma^2_{QTL}}{2pq}}$$

The genotype frequencies p and q can be determined via the genotype frequencies. To do that, we first have to derive the genotypes from the alleles.

```
library(dplyr)
tbl_qtl_geno <- tbl_qtl %>%
  mutate(Genotype = L1_pat + L1_mat - 2L)
head(tbl_qtl_geno)
```

```
## # A tibble: 6 x 5
##      ID L1_pat L1_mat Phenotype Genotype
##   <int>  <int>  <int>     <dbl>    <int>
## 1    46      2      1    -1.16         1
## 2    47      1      1    -0.780        0
## 3    48      1      2     0.345        1
## 4    49      2      1    -1.86         1
## 5    50      1      1     0.539        0
## 6    51      2      2    -1.16         2
```

The genotype frequencies are computed with

```
(tbl_qtl_freq <- tbl_qtl_geno %>%
  group_by(Genotype) %>%
  summarise(geno_count = n(),
            geno_freq = geno_count/nrow(tbl_qtl_geno)))
```

```
## # A tibble: 3 x 3
##   Genotype geno_count geno_freq
##      <int>      <int>     <dbl>
## 1        0        754     0.278
## 2        1       1376     0.507
## 3        2        584     0.215
```

The allele frequencies p and q is

```
n_geno_freq_g2g2 <- tbl_qtl_freq$geno_freq[1]
n_geno_freq_g1g2 <- tbl_qtl_freq$geno_freq[2]
n_geno_freq_g1g1 <- tbl_qtl_freq$geno_freq[3]
n_allele_freq_q <- n_geno_freq_g2g2 + 0.5 * n_geno_freq_g1g2
n_allele_freq_p <- n_geno_freq_g1g1 + 0.5 * n_geno_freq_g1g2
```

$$p = f(G_1G_1) + \frac{1}{2}\ f(G_1G_2) = 0.2151805 + \frac{1}{2} * 0.5070007 = 0.4686809$$
$$q = f(G_2G_2) + \frac{1}{2}\ f(G_1G_2) = 0.2778187 + \frac{1}{2} * 0.5070007 = 0.5313191$$

With p and q, the genotypic value a can be computed as shown above

```
n_geno_val_a <- sqrt(n_qtl_var / (2 * n_allele_freq_p * n_allele_freq_q))
```

$$a = \sqrt{\frac{\sigma^2_{QTL}}{2pq}} = \sqrt{\frac{0.2287202}{2 * 0.4686809 * 0.5313191}} = 0.6776741$$

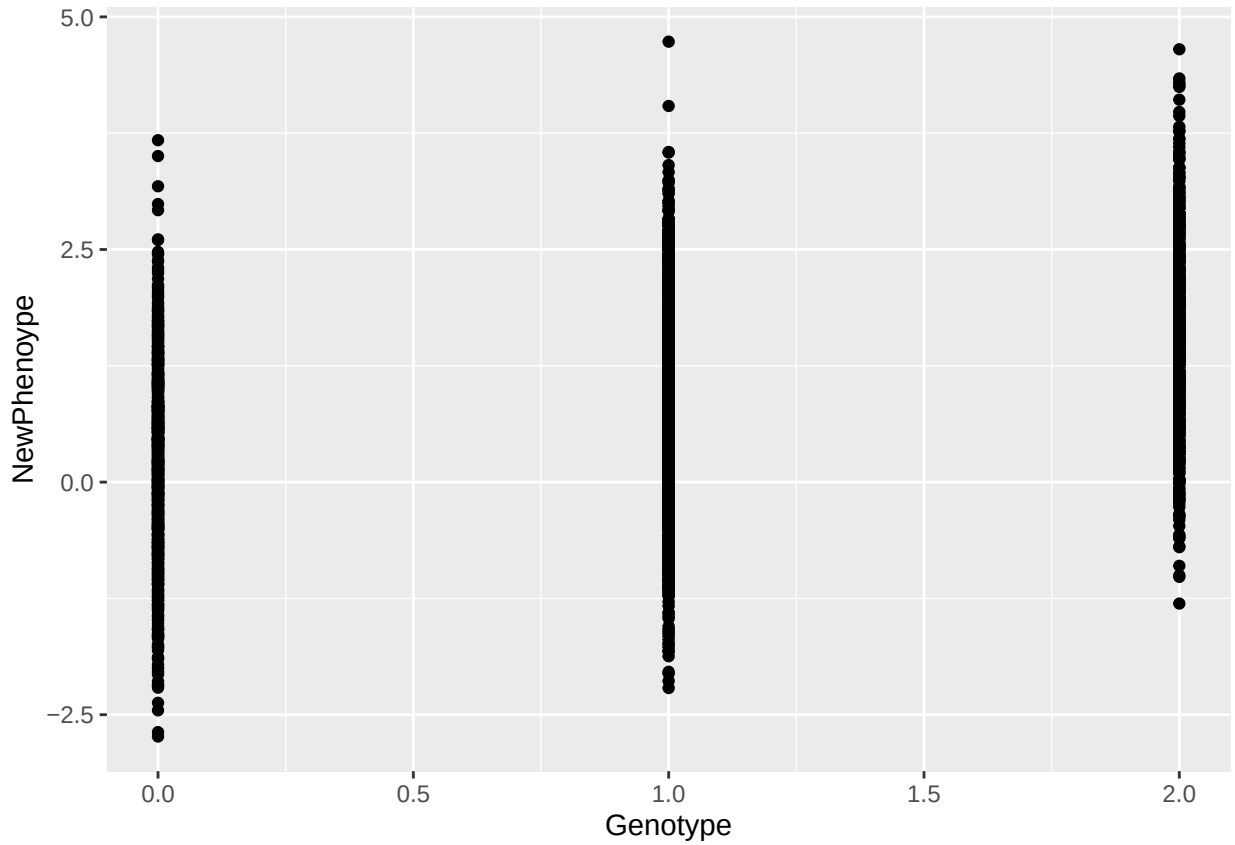
Depending on the genotype, we can now add the computed genotypic value a to the phenotype.

```
tbl_qtl_geno <- tbl_qtl_geno %>%
  mutate(NewPhenoype = Phenotype + n_geno_val_a * Genotype)
head(tbl_qtl_geno)
```

```
## # A tibble: 6 x 6
##      ID L1_pat L1_mat Phenotype Genotype NewPhenoype
##   <int>  <int>  <int>     <dbl>    <int>       <dbl>
## 1    46      2      1    -1.16         1      -0.484
## 2    47      1      1    -0.780        0      -0.780
## 3    48      1      2     0.345        1       1.02
## 4    49      2      1    -1.86         1      -1.19
## 5    50      1      1     0.539        0       0.539
## 6    51      2      2    -1.16         2       0.197
```

Use a scatterplot to show the different genotypes

```
library(ggplot2)
p <- ggplot(data = tbl_qtl_geno, aes(x = Genotype, y = NewPhenoype)) +
  geom_point()
p
```



As a check, we can compute the regression of the new phenotypes on the genotypes

```
lm_new_phen <- lm(formula = NewPhenoype ~ Genotype, data = tbl_qtl_geno)
(sry_new_phen <- summary(lm_new_phen))
```

```
##
## Call:
## lm(formula = NewPhenoype ~ Genotype, data = tbl_qtl_geno)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0994 -0.6863  0.0252  0.6795  3.8475
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.19774    0.03237   6.109 1.15e-09 ***
## Genotype     0.68870    0.02768  24.883  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.008 on 2712 degrees of freedom
## Multiple R-squared:  0.1859, Adjusted R-squared:  0.1856
## F-statistic: 619.1 on 1 and 2712 DF,  p-value: < 2.2e-16
```

The coefficient for the genotype is about what we computed as the genotypic value a. The deviation results from the fact that the original phenotypes already showed a very small effect of the genotype on the original phenotype.

The regression line can be added to the plot

```
p <- p + geom_abline(slope = sry_new_phen$coefficients["Genotype", "Estimate"],
                     intercept = sry_new_phen$coefficients["(Intercept)", "Estimate"],
                     color = "red")
p
```