Livestock Breeding and Genomics - Solution 3

Peter von Rohr

2023-09-28

Problem 1: Genotype and Allele Frequencies

Given the dataset available from:

```
## https://charlotte-ngs.github.io/lbgfs2023/data/lbgfs2023_lbg_ex03.csv
```

The dataset can be read using the following command

```
readr::read_delim("https://charlotte-ngs.github.io/lbgfs2023/data/lbgfs2023_lbg_ex03.csv",
delim = ",")
```

In the above dataset, genotypes are encoded as follows

LocusG	LocusH	Code
G_2G_2	H_2H_2	0
G_1G_2	H_1H_2	1
G_1G_1	H_1H_1	2

Your Tasks

- compute genotype frequencies
- compute allele frequencies

Solution

- Read the dataset and assign it to a tibble or dataframe

```
s_lbg_ex03_p01_data_url <- "https://charlotte-ngs.github.io/lbgfs2023/data/lbgfs2023_lbg_ex03.csv"
tbl_lbe_ex03 <- readr::read_delim(s_lbg_ex03_p01_data_url, delim = ",")
tbl_lbe_ex03
```

```
## # A tibble: 75 x 3
##    Animal LocusG LocusH
##     <dbl>  <dbl>  <dbl>
##  1      1      0      1
##  2      2      1      1
##  3      3      1      1
##  4      4      1      1
```

```
##  5      5      1      1
##  6      6      1      1
##  7      7      1      1
##  8      8      1      2
##  9      9      1      2
## 10     10      0      1
## # i 65 more rows
```

- Compute genotype frequencies using either the function `table()` or a `dplyr` pipeline The solution with `table()` results in counts which can be converted to frequencies

```
n_nr_total_animals <- nrow(tbl_lbe_ex03)
vec_freq_table_LocusG <- table(tbl_lbe_ex03$LocusG)
round(vec_freq_table_LocusG / n_nr_total_animals, digits = 3)
```

```
##
##     0     1     2
## 0.213 0.520 0.267
```

The same for locus H

```
vec_freq_table_LocusH <- table(tbl_lbe_ex03$LocusH)
round(vec_freq_table_LocusH / n_nr_total_animals, digits = 3)
```

```
##
##     0     1     2
## 0.133 0.533 0.333
```

The second solution is to use `dplyr`

```
library(dplyr)
tbl_lbe_ex03 %>%
  select(LocusG) %>%
  group_by(LocusG) %>%
  summarise(genotype_frequency = n() / n_nr_total_animals)
```

```
## # A tibble: 3 x 2
##   LocusG genotype_frequency
##    <dbl>              <dbl>
## 1      0              0.213
## 2      1              0.52
## 3      2              0.267
```

Similarly for Locus H

```
tbl_lbe_ex03 %>%
  select(LocusH) %>%
  group_by(LocusH) %>%
  summarise(genotype_frequency = n() / n_nr_total_animals)
```

```
## # A tibble: 3 x 2
##   LocusH genotype_frequency
##    <dbl>              <dbl>
## 1      0              0.133
## 2      1              0.533
## 3      2              0.333
```

Problem 2: Check for Hardy-Weinberg Equilibrium

Use the dataset from Problem 1 and check for Hardy-Weinberg equilibrium at both loci using a χ^2 test.

Solution

- Read the data, as shown above

```
s_lbg_ex03_p01_data_url <- "https://charlotte-ngs.github.io/lbgfs2023/data/lbgfs2023_lbg_ex03.csv"
tbl_lbe_ex03 <- readr::read_delim(s_lbg_ex03_p01_data_url, delim = ",")
```

- Genotype frequencies: Because of the used genotype encoding, the numbers can be interpreted as counts of the favorable alleles G_1 and H_1. So the total number of favorable alleles for both loci is given by

```
sum(tbl_lbe_ex03$LocusG)
```

```
## [1] 79
```

for Locus G and for Locus H

```
sum(tbl_lbe_ex03$LocusH)
```

```
## [1] 90
```

To get to the allele frequencies p at both loci, we have to divide these sums by the total number of alleles which is two times the number of animals in the dataset. Hence the allele frequencies are

```
n_nr_alleles_total <- 2 * n_nr_total_animals
round(sum(tbl_lbe_ex03$LocusG) / n_nr_alleles_total, digits = 3)
```

```
## [1] 0.527
```

for Locus G and

```
round(sum(tbl_lbe_ex03$LocusH) / n_nr_alleles_total, digits = 3)
```

```
## [1] 0.6
```

These frequencies can also be computed by taking the mean of each of the genotype columns and deviding them by 2

```r
round(mean(tbl_lbe_ex03$LocusG)/2, digits = 3)
```

```
## [1] 0.527
```

and analogeously

```r
round(mean(tbl_lbe_ex03$LocusH)/2, digits = 3)
```

```
## [1] 0.6
```

Hence the allele frequencies are

```r
n_allele_freq_p_locus_G <- mean(tbl_lbe_ex03$LocusG)/2
n_allele_freq_q_locus_G <- 1-n_allele_freq_p_locus_G
n_allele_freq_p_locus_H <- mean(tbl_lbe_ex03$LocusH)/2
n_allele_freq_q_locus_H <- 1-n_allele_freq_p_locus_H
```

The genotype frequencies according to Hardy-Weinberg are given by

```r
vec_geno_freq_locus_G_hw <- c(n_allele_freq_q_locus_G^2,
                              2*n_allele_freq_q_locus_G * n_allele_freq_p_locus_G,
                              n_allele_freq_p_locus_G^2)
round(vec_geno_freq_locus_G_hw, digits = 3)
```

```
## [1] 0.224 0.499 0.277
```

and the same for locus H

```r
vec_geno_freq_locus_H_hw <- c(n_allele_freq_q_locus_H^2,
                              2*n_allele_freq_q_locus_H * n_allele_freq_p_locus_H,
                              n_allele_freq_p_locus_H^2)
round(vec_geno_freq_locus_H_hw, digits = 3)
```

```
## [1] 0.16 0.48 0.36
```

- Run χ^2 test with then genotype counts from the data

```r
chisq.test(table(tbl_lbe_ex03$LocusG), p = vec_geno_freq_locus_G_hw)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(tbl_lbe_ex03$LocusG)
## X-squared = 0.13846, df = 2, p-value = 0.9331
```

for locus H

```r
chisq.test(table(tbl_lbe_ex03$LocusH), p = vec_geno_freq_locus_H_hw)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(tbl_lbe_ex03$LocusH)
## X-squared = 0.92593, df = 2, p-value = 0.6294
```

Although there are differences in the distribution of genotype frequencies, the χ^2 test does not give us a result in the test-statistic that would suggest that there is a detectable deviation from the Hary-Weinberg equilibrium.