Homework #5

Kate Harline feat. T the Bone Always Wrightsman

3/12/2019

Problem 1

A

A causal polymorphism ($A_1 - > A_2$) is a change in the DNA (δY) at a given locus which leads to a change in the phenotype under given conditions (Z).

This can be represented as: $A_1 - > A_2 \implies \delta Y | Z$

B

The uncertainty in X ϵ $N(0, \sigma_\epsilon^2)$

Problem 2

```
## PART A ##
set.seed(666)
choose_genotype <- function(int){

  switch (int, 'A1A1', 'A1A2', 'A1A2', 'A2A2')
}

sim_genotypes <- function(num_rows, num_cols){
  genotypes <- matrix(0, nrow = num_rows, ncol = num_cols)
  # randomly append 1 thru 4

  genotypes <- apply(genotypes, c(1,2), function(x){sample(1:4, 1)})
  genotypes <- apply(genotypes, c(1,2), choose_genotype)
  return(genotypes)
}
genotypes <- sim_genotypes(400, 500)
```

Part B

```
# set up new matrix
xa_matrix <- matrix(NA, nrow = nrow(genotypes), ncol = ncol(genotypes))
# fill with x_a values
for (i in 1:ncol(genotypes)) {
  xa_matrix[,i] <- sapply(genotypes[,i], (function(x) switch (x,
    'A1A1' = -1, 'A1A2' = 0, 'A2A2' = 1
  )))
}
```

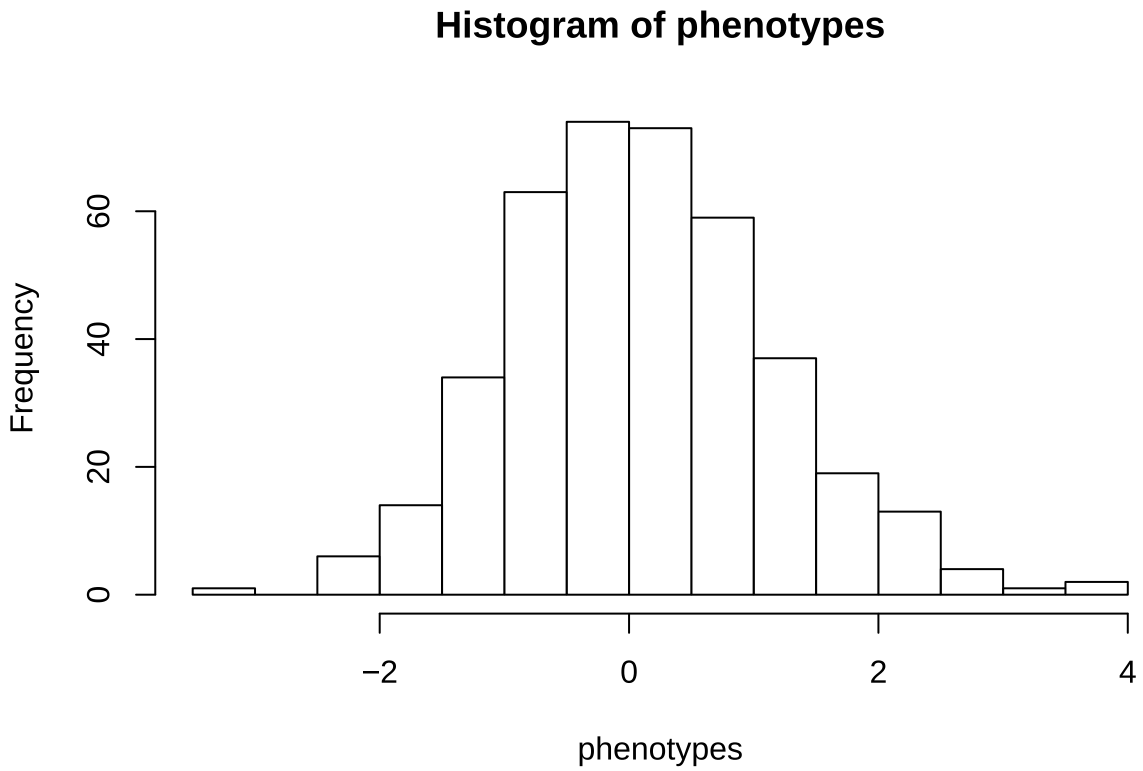
```
# then convert x_a to x_d
xd_matrix <- -2*abs(xa_matrix)+1
```

Part C

```
 #???? = 0.1, ??a,100 = 0.5, ??d,100 = ???0.2, ????2 = 1
b_u <- 0.1
b_a <- 0.5
b_d <- -0.2
sig_sq <- 1
# calculate phenotypes
phenotypes <- b_u + b_a * xa_matrix[,100] + b_d * xd_matrix[,100] + rnorm(400, 0, sqrt(sig_sq))
```

Part D

```
hist(phenotypes)
```



## Part E, F, G

```
# install.packages(MASS)
library(MASS)
# Function to calculate the pval given a set of individuals' phenotype, and genotype encodings.
pval_calculator <- function(pheno_input, xa_input, xd_input){
    n_samples <- length(xa_input)

    X_mx <- cbind(1,xa_input,xd_input)
    MLE_beta <- ginv(t(X_mx) %*% X_mx) %*% t(X_mx) %*% pheno_input
    y_hat <- X_mx %*% MLE_beta

    SSM <- sum((y_hat - mean(pheno_input))^2)
    SSE <- sum((pheno_input - y_hat)^2)

    df_M <- 2
    df_E <- n_samples - 3

    MSM <- SSM / df_M
    MSE <- SSE / df_E

    Fstatistic <- MSM / MSE

    pval <- pf(Fstatistic, df_M, df_E,lower.tail = FALSE)

    return(pval)
}

# Initialize some variables and constants
n_geno <- ncol(xa_matrix)
n_pheno <- length(phenotypes)
pvals <- c()

# Calculate and save pvals for each phenotype-genotype pair

  for (i in 1 : n_geno){
      pvals <- c(pvals, pval_calculator(phenotypes,
                                        xa_input = xa_matrix[,i],
                                        xd_input = xd_matrix[,i]))
  }
```

Part H

```
library(ggplot2)
```

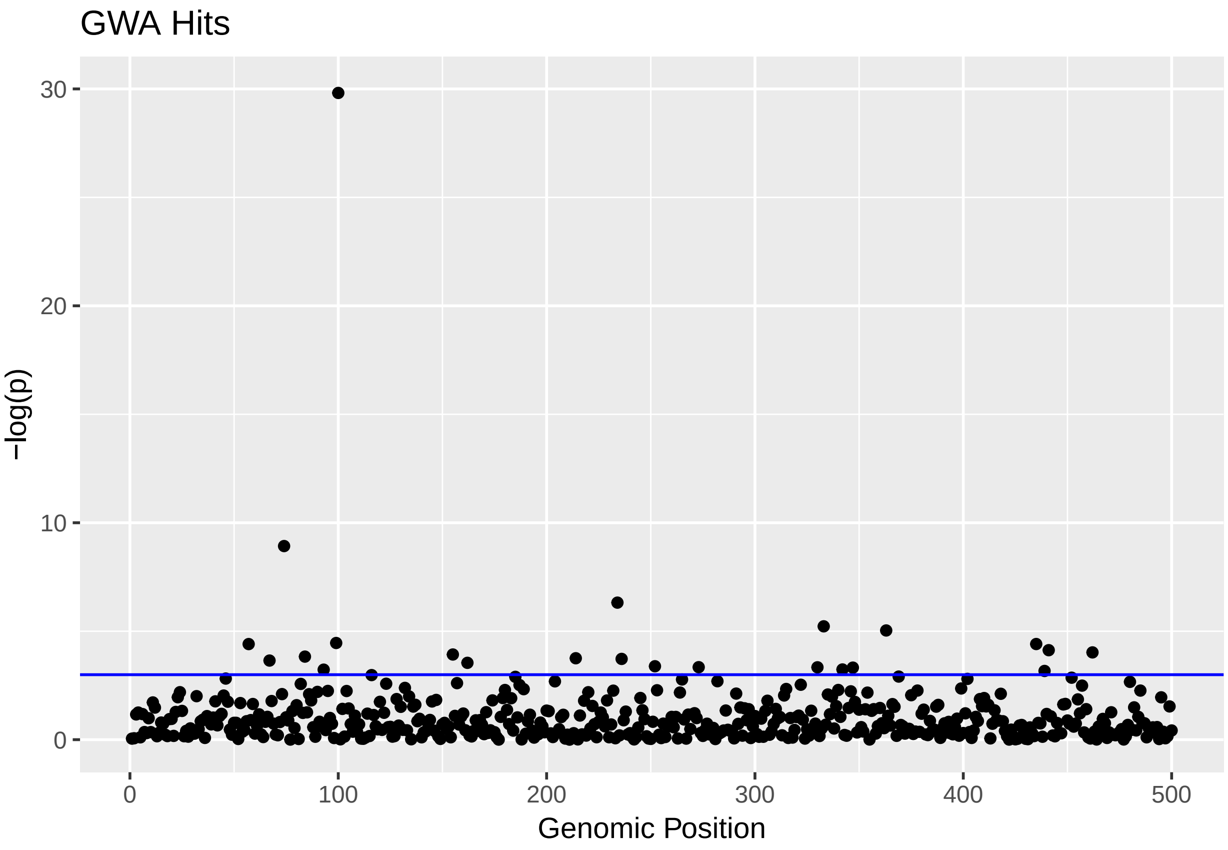
```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library(gridExtra)

neg_log_p <- -1 * log(pvals)
x <- seq(1, length(neg_log_p))
y <- neg_log_p

plot1 <-ggplot(data = data.frame(x, y), mapping = aes(x, y)) + geom_point() + geom_hline(yintercept=-log

print(plot1)
```



Part I

```
cat('Significant hits for Phenotype with type 1 error 0.05 = ', sum(pvals < 0.05), '\n')
```

```
## Significant hits for Phenotype with type 1 error 0.05 =  23
```

Part J

I rejected the H_o that a genotype was not casual 25 times. It is not surprising the null was rejected 25 times because we set $\alpha = 0.05$ which means we are permitting a Type I error of 5% and for 500 samples we randomly expect $0.05 * 500 = 25$. In our system, all of the genotypes were simulated randomly so they don't exhibit the phenomenon of linkage disequillibirum which is what usually allows us to infer causal genotypes from GWA of real data. In Linkage disequilibirum, loci that are close together are inherited together therefore their impact on the phenotype is correlated. So, even if we do not measure the precise position that is causal, we can still infer the region of the genome with which it is in linkage disquilibrium as being causal. We removed this factor by the way we simulated the data, had we structured the assignment of genotypes around loci 100, such that there was a predictably linked "haplotype" pattern, then we would be able to simulate this inference.

Problem 3

Given,

$Y = \beta_\mu + X_a * \beta_a + X_d * \beta_d + \epsilon$

If

$\beta_a = 0, \beta_d = 0$

This simplifies to

$Y = \beta_\mu + 0 + 0 + \epsilon$

Which implies that Y is independent of X_a, X_d which means that the $Cov(X_a, Y) \cap Cov(X_d, Y)$.