Midterm Problems Writeup (Please see additional RMD File on CMS for all code placed together with comments)

BTRY 6830

November 14, 2020

- ”Chunks” refer to separate parts, as given, in my RMD file.

1 Problem

1.1 Part A: Import Sample ID and BP Data (Chunk 1)

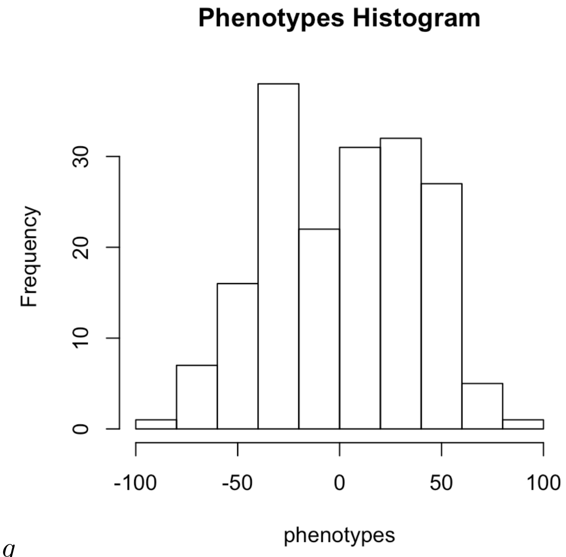
```
    > phenotypes <- read.csv("midterm2019_pheno+pop.csv", row.names =1)
```

1.2 Part B: Calculating the Number of Rows in the Phenotype File (Chunk 1)

```
    > nrow(phenotypes)
[1] 180
```

1.3 Part C: Histogram of BP Phenotypes with x axis label of phenotypes and y axis label of frequency (Chunk 1)

```
> phenotypes <- as.matrix(phenotypes)
> hist(phenotypes, main = "Phenotypes Histogram")
```



(41).*png*

1.4 Part D

From the histogram, we see that it deviates from the normal distribution because, roughly speaking, the 'bins' corresponding to the phenotype data are not concentrated towards the center of the plot. This observation not only provides information about our GWAS data, in particular that even if the phenotype data is not normally distributed, then a linear regression could still apply because we could transform the phenotype data to make it 'more' normal, but also that this deviation from the normal distribution could be explained by differences in MAF and mean phenotype in the imported data.

1.5 Part E

From this phenotype data, it would not be appropriate to use a logistic regression because, even from the histogram alone, it is clear that our data is not normally distributed, and we would therefore not be able to model our data accurately with a normal error term.

2 Problem

2.1 Part A: Import Genotype Data (Chunk 1)

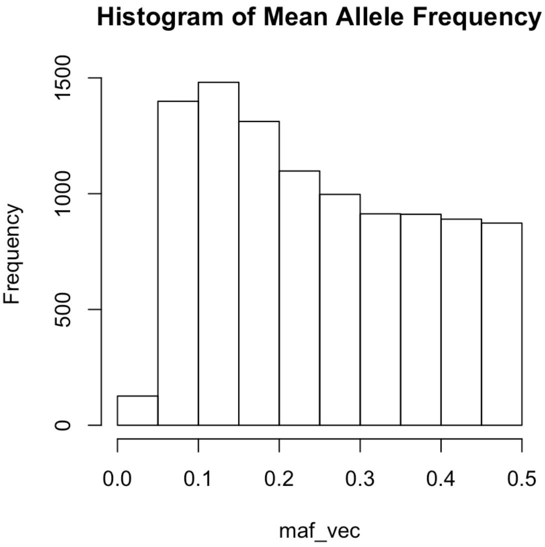
```
    > genotypes <- read.csv("midterm2019_genotypes.csv")
```

2.2 Part B: Reporting number of SNPs N (Chunk 1)

```
    > N_genotypes <- ncol(genotypes)
> N_genotypes
[1] 10000
> n_samples <- nrow(genotypes)/2
> n_samples
[1] 90
```

2.3 Part C: Calculating MAF for each SNP (Chunk 2), and the Histogram (Chunk 3)

```
> maf_calc <- function(single_geno){
+     n_geno <- length(single_geno)
+     allele_count <- sum(single_geno)
+     allele_freq <- allele_count / (2*n_geno)
+     if(allele_freq > 0.5){
+         allele_freq <- 1- allele_freq
+     }
+     return(allele_freq)
+ }
```



(46).*png*

2.4 Part D

Keeping in mind that the power is defined as the probability of rejecting the null when the alternative is true, we recall that the 'power' of a hypothesis test is, for a one-sided test, rigorously defined as

$$1 - \beta = \int_{c_\alpha}^{\infty} \Pr(T(\mathbf{x}|\theta)\mathrm{d}T(x) \ ,$$

where

$$\beta = -\int_{-\infty}^{c_\alpha} \Pr(T(\mathbf{x}|\theta)\mathrm{d}T(\mathbf{x}) \ ,$$

is the Type II Error, and $\mathcal{H}_0 = \theta - c$.

2.5 Part E

To determine how power relates to allele frequency, we first observe that low MAF would certainly impact our power in a statistical analysis, in general, because we would need to take a higher number of samples to get adequate and reliable observations. For a MAF of 0.001, we would need 100,000 samples because for this MAF, we would require at least 1000 samples to observe is 1 time. As a result, a low MAF, even lower than the 0.05 cutoff that has been discussed in lecture, would drastically impact our power, negatively.

3 Problem

3.1 Part A

3.1.1 Xa and Xd Encodings (Chunk 4)

```
# Introduce Xa and Xd encoding for genotypes
```

```
xa_matrix <- genotypes - 1
xd_matrix <- 1 - 2 * abs(xa_matrix)
```

3.1.2 Calculating p values from the X_a and X_d matrices (Chunk 5)

```
    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

     # to check if it is correct
     pval <- pf(Fstatistic, df_M, df_E,lower.tail = FALSE)

     return(pval)
}
```

3.1.3 Creating the p values vector (Chunk 6)

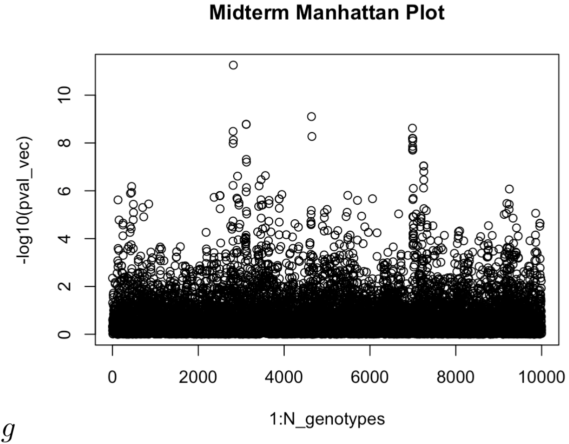
```
    pval_vec <- c()
for (i in 1 : N_genotypes){

     #cat("Testing Gentoype = ", i, "\n")
     pval_vec[i] <- pval_calculator(phenotypes, xa_input = xa_matrix[,i], xd_input = xd_matrix[,i])

}
```

3.1.4 Output of p val vector

```
  > pval_vec
    [1] 4.669639e-01 4.507968e-03 6.930814e-01 4.842653e-02 2.438988e-01
   [6] 4.707244e-01 5.571409e-01 9.418077e-01 2.727772e-01 9.721629e-01
  [11] 7.248203e-01 4.866904e-01 3.260295e-01 7.712414e-01 6.117156e-01
  [16] 2.162814e-01 2.279255e-02 8.229016e-01 5.773584e-01 2.889020e-02
  ...
```

3.2 Part B: Manhattan Plot (Chunk 7)

```
   > plot(1: N_genotypes, -log10(pval_vec), main = "Midterm Manhattan Plot")
```

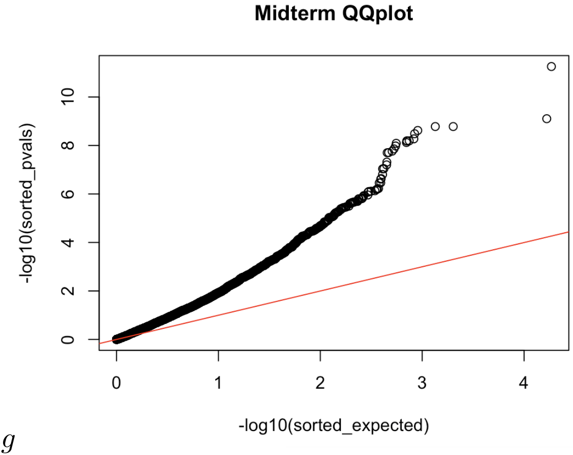


(43).*png*

4 Problem

4.1 Part A: QQ Plot (Chunk 7)

```
sorted_pvals <- sort(pval_vec, decreasing = FALSE)
sorted_expected <- sort(runif(N_genotypes), decreasing = FALSE)
    plot(-log10(sorted_expected), -log10(sorted_pvals), main = "Midterm QQplot")
abline(a = 0, b = 1, col = "red")
```



(45).*png*

4.2 Part B

From the QQ plot, I could explain to a collaborator that I do not think that the appropriate analysis had been applied to this GWAS data, because as mentioned several times in lecture and in lab, the data departs too quickly off the line $y = x$ in the QQ plot that we have provided. Therefore, if my explanation were to be correct, from the shape of the QQ plot alone, we are able to conclude that the particular GWAS analysis that we carried out were not appropriate. Furthermore, we would expect that the QQ plot would have this shape if we did not introduce covariates, and pursue covariate analysis, as we calculated the p values from our GWAS data.

5 Problem

From the hint, we can immediately extract the substring that would show us the two separate populations from the phenotype file, by first observing that the two populations, from the phenotype file, are given, respectively, from indices 1 to 89, and then from index 90 to 181. Therefore, $n_1 = 89$, and $n_2 = 181 - 90 = 91$. At the indices that we have given in the first column of the phenotypes file, we see a difference in naming of the individuals in each population,

```
HG00384
```

and

```
NA20502
```

which clearly shows tat $n_1 + n_2 = 89 + 91 = 180$, which is exactly the vlaue that we reported for the total sample size in **1B**, after importing the phenotypes file.

5.1 Part B

Intuitively, it would still be possible for us to show that the individuals represent 2 distinct populations because when we perform a PCA analysis, the procedure of arranging axes orthogonal to each other, and replotting individuals along the principal component with the largest variance, would visually distinguish the populations from one another.

6 Problem

6.1 Part A

6.1.1 R Covariates Code From Lab 10 (Chunk 9)

```
pval_calculator_lab10 <- function(pheno_input, xa_input, xd_input, z_input){
n_samples <- length(xa_input)

# Set up random variables for null (Z_mx) and with genotypes (XZ_mx)
Z_mx <- cbind(1,z_input)                                                # H0 (w/
XZ_mx <- cbind(1,xa_input,xd_input,z_input)                             # w/ geno

# Calculate MLE betas for both null model and model with genotypes and covariates
MLE_beta_theta0 <- ginv(t(Z_mx)  %*% Z_mx)  %*% t(Z_mx)  %*% pheno_input       # H0 (w/
MLE_beta_theta1 <- ginv(t(XZ_mx) %*% XZ_mx) %*% t(XZ_mx) %*% pheno_input       # w/ geno

# Get Y estimates using the betas calculated above to give each hypothesis its best chance
y_hat_theta0 <- Z_mx  %*% MLE_beta_theta0                               # H0 (w/
y_hat_theta1 <- XZ_mx %*% MLE_beta_theta1                               # w/ geno

# Get the variance between the true phenotype values and our estimates under each hypothesis
SSE_theta0 <- sum((pheno_input - y_hat_theta0)^2)                       # H0 (w/
SSE_theta1 <- sum((pheno_input - y_hat_theta1)^2)                       # w/ geno

# Set degrees of freedom
df_M <- 2
```

```
    df_E <- n_samples - 3

    # Put together calculated terms to get Fstatistic
    Fstatistic <- ((SSE_theta0-SSE_theta1)/df_M) / (SSE_theta1/df_E)

    # Determine pval of the Fstatistic
    pval <- pf(Fstatistic, df_M, df_E,lower.tail = FALSE)
    return(pval)
 }
```

6.1.2 Correctly Representing the covariate factors -1 and $+1$ for the two populations with $n_1 = 89$ and $n_2 = 91$ individuals (Chunk 10)

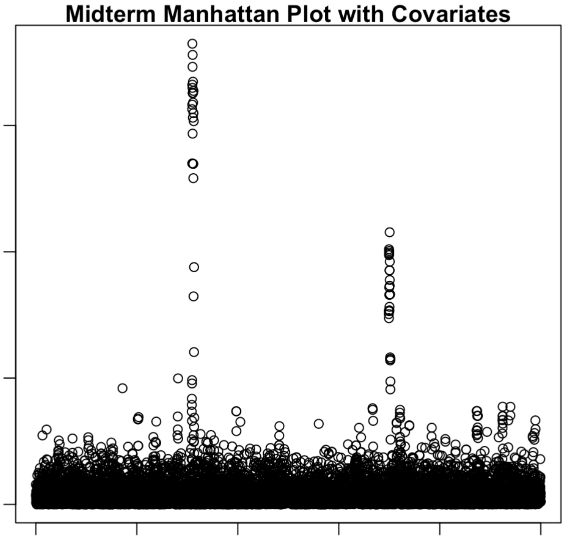
```
  > x_1 <- sample(c(-1,1), n_1 , replace = T, prob = c(1,0))
> x_2 <- sample(c(0,1), 180-n_1, replace = T , prob = c(1,0))
> x_3 <- c(x_1,x_2)
> y_1 <- sample(c(0,1), 90 , replace = T , prob = c(1,0))
> y_2 <- sample(c(1,0), 90 , replace = T , prob = c(1,0))
> y_3 <- c(y_1 , y_2)
> Z_value_n1n2 <- cbind(x_3, y_3)
```

6.1.3 Output of p val covariates vector

```
    > pval_vec_covariates
  [1] 0.163293084 0.408027269 0.502761903 0.429972434 0.829937053 0.641990437 0.863682321 0.3168795
  [13] 0.301500320 0.644597410 0.756942246 0.573013153 0.657339124 0.999345184 0.359379034 0.182934
  [25] 0.692556393 0.615263873 0.379494172 0.960257843 0.071114159 0.303512699 0.169843755 0.597313
  ...
```

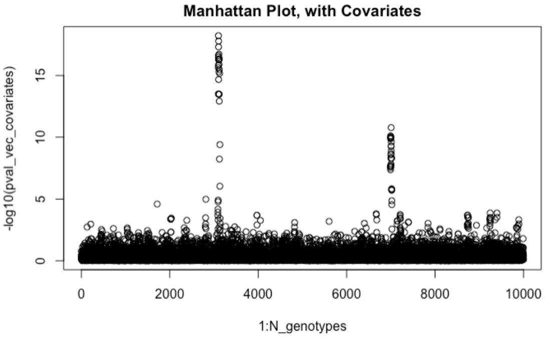
6.2 Part B: Manhattan Plot (Chunk 11)

```
   > plot(1: N_genotypes, -log10(pval_vec_covariates), main = "Midterm Manhattan Plot with Covariat
```



(48)*.png*

Or, with more appropriately labeled axes,



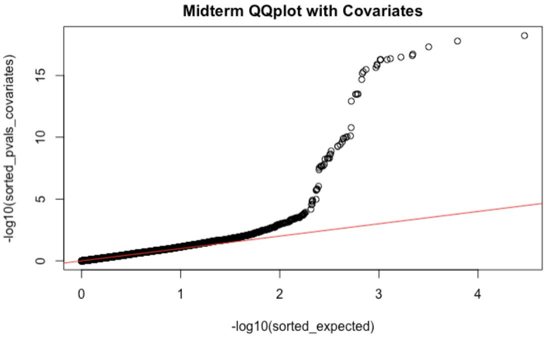
(51).*png*

7 Problem

7.1 Part A: QQ Plot (Chunks 10/11)

```
> sorted_pvals_covariates <- sort(pval_vec_covariates, decreasing = FALSE)
> sorted_expected <- sort(runif(N_genotypes), decreasing = FALSE)
    > plot(-log10(sorted_expected), -log10(sorted_pvals_covariates), main = "Midterm QQplot with C
> abline(a = 0, b = 1, col = "red")
```

With slightly readjusted axes with better names,



(52).*png*

7.2 Part B

In this situation, I would explain to my collaborator that the analysis that we have applied, with the p values from the covariates accounting for n_1 and n_2 individuals in separate populations, helped us obtain data that is more realistic because, in contrast to the plot in **4A**, the QQ plot above stays along the line $y = x$ for more p values. From the QQ Plot, we see that our data would present a GWAS analysis that is more reliable because from the p values, we would not only be more confident in our power, but also in our ability to reject the null hypothesis when the alternative is true. The observed shape captures how introducing covariates can help improve our GWAS analysis.

8 Problem

8.1 Part A: Reporting the Appropriate p-value Cut Off, from the Bonferroni Correction, with the formula (Chunk 13)

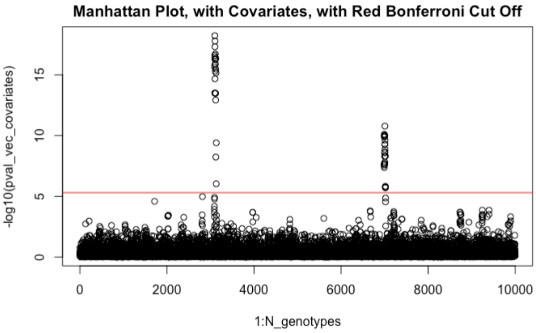
```
    cutoff <- 0.05/N_genotypes;
    which(pval_vec_covariates < 0.05/N_genotypes)
 [1] 3096 3097 3098 3099 3100 3102 3103 3104 3105 3107 3108 3109 3110 3111 3112 3114 3115 3117
[19] 3119 3121 3122 3123 3127 3132 3133 6990 6991 6992 6993 6994 6995 6996 6997 6998 6999 7000
[37] 7001 7002 7004 7005 7006 7007 7008 7009 7010 7011 7013 7014 7017 7018 7019 7020
```

8.2 Part B: Reporting Separate Peaks of the Manhattan plot with the Bonferroni Cut Off (Chunk 11)

From our Manhattan Plot in **6B**, we report 2 peaks that we can determine by enforcing the criterion from the Bonferroni correction, in which we calculate the -log base 10 transformation of the type I error divided by the number of genotypes. With this criterion, the 2 significant peaks in the Manhattan Plot below lie above the red line, with the code and plot pasted below.

```
     > 0.05/N_genotypes
[1] 5e-06
    > plot(1:N_genotypes, -log10(pval_vec_covariates), main
    = "Covariates Plot with Red Bonferroni Cut Off" )
> abline(h = -log10(5e-06), col = "red")
```

Again with appropriate axis names,



(53)*.png*

8.3 Part C

I would explain that from the 2 peaks that we have identified in the Manhattan plot above, we would reasonably expect that each peak indicate at least 1 causal genotype. As a result, because the peaks are nearly equally spaced in the Manhattan plot, and also that each peak could represent around half of the p- values that are an impact of genotypes that are in LD, we could try to estimate the number of causal genotypes from the list of p values in **8A**, but we would never be absolutely sure of the exact number of causal genotypes corresponding to each of these 2 peaks, rather than identifying at least one causal genotype for each peak.

8.4 Part D

While talking with my collaborator, I would emphasize that the 2 peaks, both of which correspond to the significant SNPs N_{maxSNP} and N'_{maxSNP}, that we have reported likely **only** indicate the positions of causal polymorphisms, because as previously mentioned, although we introduced the covariates -1 and $+1$ to improve our GWAS analysis, the populations with n_1 and n_2 individuals could still differ in terms of their MAF and mean phenotype. In turn, we are not exactly sure of the location of causal polymorphisms because of these factors, which is one of the reasons as to why it is helpful to apply covariates to $\mathcal{H}_0$.

9 Problem

9.1 Part A: Determining the Most Significant SNP in Each Peak (Chunk 14)

From the 2 peaks that we have identified, we can list the p value for the most significant SNP, as well as the number of the SNP, by determining which p values from **8A** are the largest. In particular, we numerically report that the largest SNP for the **first** peak in the Manhattan plot, which corresponds to p values between $N_1 = 3096$ and $N_2 = 3133$, is

```
[1] 9.324385e-07
```

whose SNP number N_{maxSNP}, is $N_1 = 3096 < N_{\text{maxSNP}} = 3132 < 3133 = N_2$, from the output, in Chunk 14,

```
   2.261411e-16 3.182525e-14 1.501308e-16 2.429347e-17 5.790787e-19 3.232585e-17 4.754651e-18 2.10
```

On the other hand, for the **second** peak that we identified in the Manhattan plot that was above the Bonferroni cut off, the largest SNP that we observed, from $N_1' = 6990$ to $N_2' = 7020$, was

```
   [1]2.005692e-06
```

whose SNP number N'_{maxSNP}, is $N_1' = 6990 < N'_{\text{maxSNP}} = 7018 < 7020 = N_2'$, from the output in Chunk 14,

```
   2.930976e-08 2.176148e-08 2.176148e-08 4.246501e-08 7.924058e-11 1.360782e-10 1.069234e-10 1.21
```

9.2 Part B

We could expect that the most significant SNP is not necessarily the closest to a causal polymorphism because of inherent differences between populations in our GWAS, in terms of mean phenotype and MAF. As a general fact, the SNP density within a genome is not homogeneous, and in our case, the locations of the most significant SNPs, due to potential differences in the MAF and mean phenotype for each population, could, by coincidence, not necessarily be as close to a causal polymorphism as one would expect.

9.3 Part C: Calculating the Correlation Between Nearest Neighbor SNPs to the Most Significant SNP for each Peak in the Manhattan Plot (Chunk 15)

For the most significant SNP in each peak, we can determine the correlation with the closest SNP, with respect to the X_a encoding, by making use of the genotype numbers N_{maxSNP} and N'_{maxSNP} from **A**. With these genotype numbers, it is possible to calculate the correlation of the closest SNP to the most significant SNP, for each peak, respectively at the genotype values N_{maxSNP} and N'_{maxSNP}, by first observing that the correlation between these SNPs and the ones that are closest to both of them are located at positions 3133, with respect to N_{maxSNP} located at 3132, and either at positions 7017 or 7019, with respect to N'_{maxSNP} located at 7018.

Now, we can look up these genotype numbers in the X_a matrix, which will give us the appropriate X_a encoding, which we can to calculate the correlation with respect to, by first determining that section of the X_a matrix. For N_{maxSNP}, the output

```
   > y_1 = xa_matrix[,3132]
> y_2 = xa_matrix[,3133]
```

In R, the correlations are directly,

```
    > c(cor(xa_matrix[,3132], xa_matrix[,3133]))
[1] 0.7500916
> c(cor(xa_matrix[,7018], xa_matrix[,7017]))
[1] 0.9949122
> c(cor(xa_matrix[,7018], xa_matrix[,7019]))
[1] 0.9706856
```

I think that these are the right values, but below I have indicated another way that I tried if you think that something is wrong.

9.3.1 Incorrect Matlab Attempt at Correlation

After using the Matlab function rcorr, that we have a matrix whose determinant we can take to determine the correlation. Computing the determinant of this corrleation matrix indeed gives a nonero, positive, value, because storing the vectors under the same name in Matlab gives a correlation value of

```
>> y = corr(y_1 , y_2)

y =

    0.7501
```

after substituting in

```
>> y_1
```

and also

```
 >> y_2
```

so that we can pass these 2 vectors as parameters to rcorr. On the other hand, for N'_{maxSNP}, the output

```
    > x_1 <- xa_matrix[,7018]
> x_2 <- xa_matrix[, 7017]
```

or the output

```
    > x_3 <- xa_matrix[,7019]
```

gives us both possibilities in the correlation between the SNPs that we would want to calculate. Again, using the Matlab function rcorr gives a matrix whose determinant we can take to determine the correlation. Computing the determinant of this corrleation matrix indeed gives a nonero, positive, value,

```
    >> x_1
```

and

```
    >> x_2
```

gives a correlation value of

```
    >> y = corr(x_1 , x_2)

y =

    0.1387
```

In the second case,

```
>> x_3
>> y = corr(x_1 , x_3)

y =

    0.1479
```

See bottom [1]

9.4 Part D

From **C**, it makes sense that the correlations are not very close to 0 because in a real GWAS, we would expect that Linkage Disequilibrium exists, because if we measure genotypes that are close together, namely on the same chromosome, these genotypes would be correlated with the causal genotype whose location we are trying to find on the genome. Therefore, this nonzero correlation reflects one of the assumptions in our GWAS, namely that genotypes that are close to each other in the genome are correlated.

10 Problem

10.1 Part A

To provide a rigorous definition, we recall that a causal mutation, or polymorphism, is taken as a position in the genome where an experimental manipulation of DNA produces an effect on the phenotype, under specified conditions. Symbolically, this can be represented with $A_1 \to A_2 \Rightarrow \Delta\bar{Y}|Z$, or even as $A_1 \to A_2 \Rightarrow \Delta\hat{Y}$, where A_1 and A_2 represent alleles, Y a phenotype, and Z a set of specifiable conditions. Rigorously, a causal polymorphism is then defined as a location (locus) in the genome where there are at least 2 alleles, in which experimentally switching out one allele for another, under specifiable conditions, leads to a change in phenotype.

10.2 Part B

In an ideal experiment, we would expect to identify a causal polymorphism by measuring the phenotype of interest for N genotypes of n individuals, and with these measurements, we would also perform N hypothesis tests to determine whether we would reject the null hypothesis. As we are carrying out the GWAS analysis, we would be interested in determining how accurately rejecting the null hypothesis would help locate a section of the genome that contains a causal polymorphism.

10.3 Part C

To rigorously define a p-value, we recall that this value measures the probability of obtaining a value of a statistic $T(\mathbf{x})$, or more extreme, the conditional on the null $\mathcal{H}_0$ being true as

$$\text{pval} = \Pr(|T(\mathbf{x})| \geq t|\mathcal{H}_0 : \theta = c) \ ,$$

where

[1] Dr. Mezey said either way that I get the output is fine, I just got it working in Matlab only. I hope that these numbers are fine.

$$\text{pval}(T(x)) : T(x) \rightarrow [0,1] \ .$$

In the case of a one-sided test, as defined in lecture, the p-value is more technically defined as

$$\text{pval}(T(\mathbf{x}) = \int_{T(\mathbf{x})}^{\infty} \Pr(T(\mathbf{x})|\mathcal{H}_0 : \theta = c) \mathrm{d}T(\mathbf{x}) \ ,$$

which, discretely, is the analogous to

$$\text{pval}(T(\mathbf{x})) = \sum_{T(\mathbf{x})}^{\max T(\mathbf{X})} \Pr(T(\mathbf{x})|\theta = c) \ .$$

10.4 Part D

Three reasons as to why there could be no causal polymorphism in the genomic location of the polymorphism that we analyzed are, first, that there could be several populations in our samples that could have different allele frequencies; second, that these populations could also differ in terms of the mean phenotype that we are interested in measuring, and third, that the populations could also differ in MAF, on a subset of measured genotypes. These 3 reasons, and several more that are provided on **1B** from the HW 6 key, demonstrate how biological false positive can influence our GWAS analysis, and even more importantly, how we must account for different biological false positives with covariates.

11 References

- https://www.biostars.org/p/153891/
- Lecture 16 video and slides
- Lecture 18 video and slides
- HW 5 Key
- https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues
- Lecture 12 Slides
- https://en.wikipedia.org/wiki/Power(statistics)
- https://stackoverflow.com/questions/22906804/matrix-expression-causes-error-requires-numeric-complex-matrix-vector-arguments
- https://www.rdocumentation.org/packages/base/versions/3.5.3/topics/substr
-