Midterm

Kate Harline

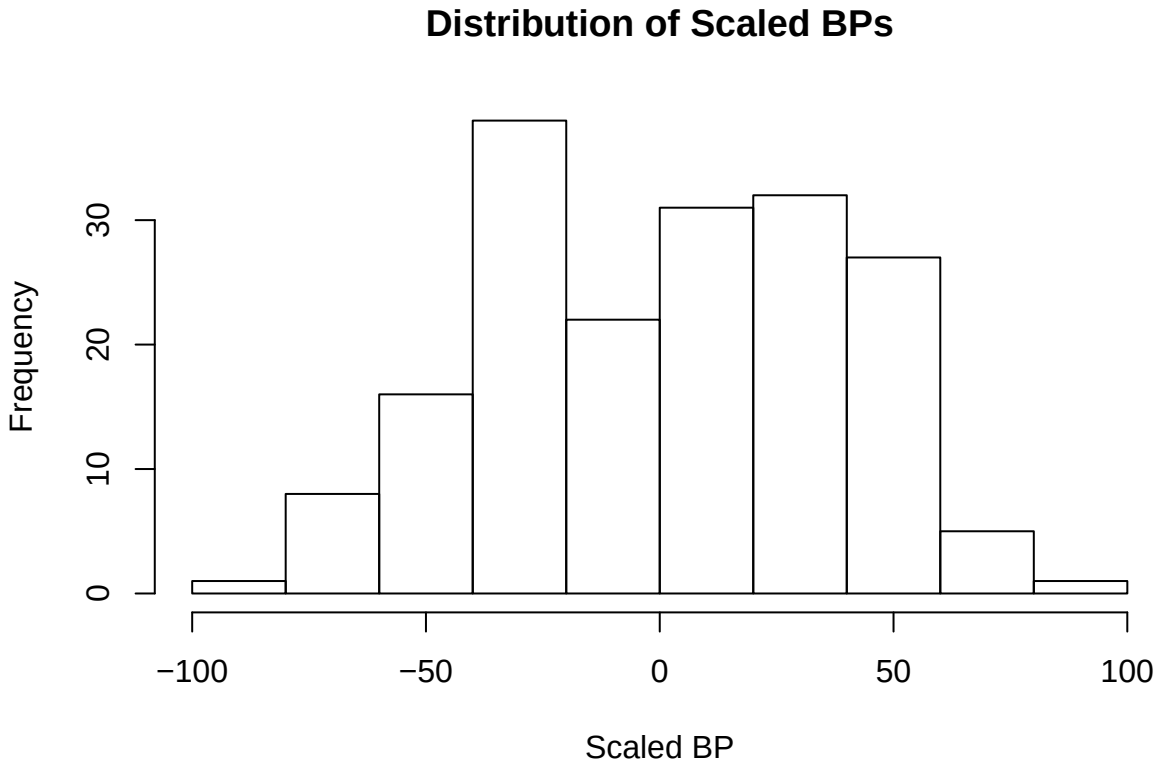
4/12/2019

1

```
# a import the sample Id and BP data
phenos <- read.table("./midterm2019_pheno+pop.csv", header = F, sep = ',')

# b calculate and report sample size n
n <- length(phenos[,1])
cat('There are n = ', n, 'samples')
```

```
## There are n =  181 samples
```

```
# c plot a histogram of bp phenotypes
bp_hist <- hist(phenos[,2], main = 'Distribution of Scaled BPs', xlab = 'Scaled BP')
```



d

Here there are approximately two peaks straddling the mean as opposed to one peak in the middle like in a typical normal distribution; this deviation is likely caused by the underlying population struction that is confounding the assumption: the phentoypes are normally distributed. A linear regression could still be appropriate because the BP data is a continuous variable and we can remove the noise from the data by incorporating the population structure as a covariate.

e

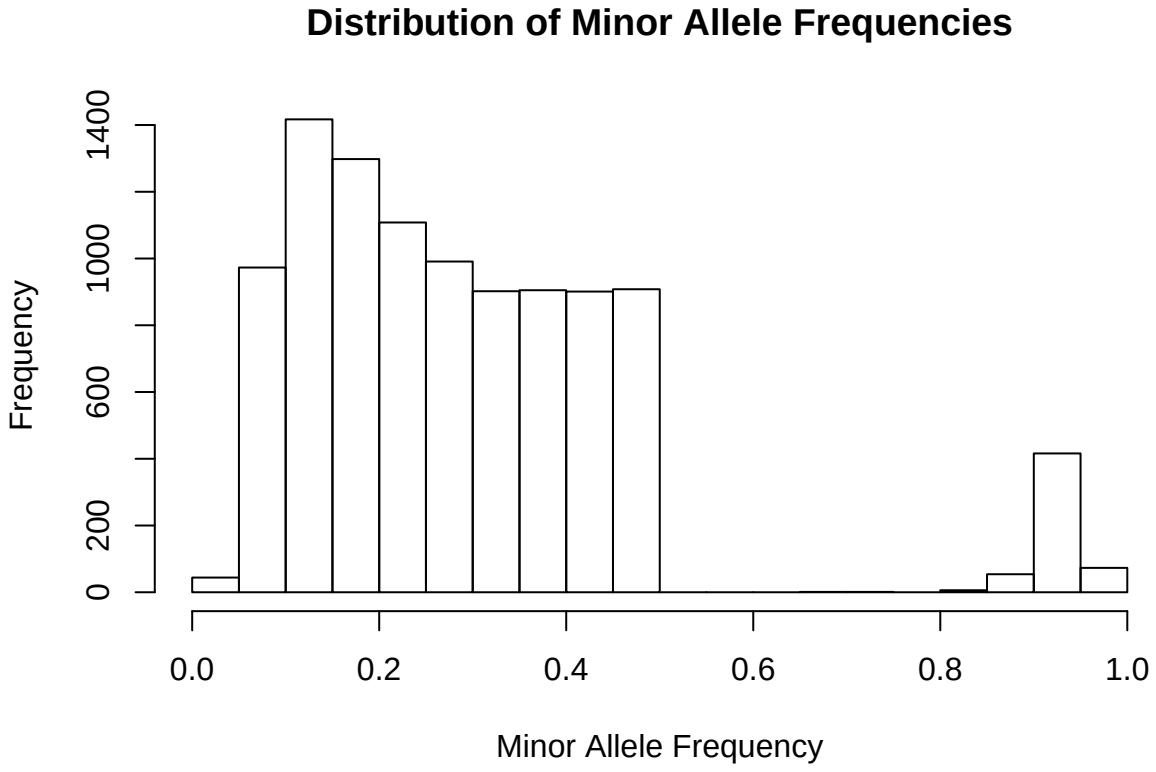
Logistic models are better for discrete phenotype data of two states because the error term follows a Bernoulli distribution $\epsilon \sim B(p)$, rather than linear models where $\epsilon \sim N(\mu, \sigma_\epsilon^2)$ which better accounts for the possible values of continuous phenotype data.

2

```
# a import the genotype data
genos <- read.table("./midterm2019_genotypes_v2.csv", header = F, sep = ',')

# b calculate and report the number of SNPs, N
N <- ncol(genos)
cat('There are N = ', N, ' SNPs.')
```

```
## There are N =  9998  SNPs.
```

```
# c calculate the MAF for each SNP -- we do not know if this is 0 or 2,
xa_mx <- matrix(data = NA, nrow = nrow(genos), ncol = ncol(genos))
# for each SNP
for(i in 1:ncol(genos)){
  # remove hets
  genos_to_count<-which(genos[,i] != 1)
  geno_count <- table(genos[genos_to_count,i])
  minor_allele <- names(geno_count[geno_count == min(geno_count)])
  # minor allele homo is 2
  xa_mx[,i] <- genos[,i] - 1
  # minor allele homo is 0
  if(0 %in% minor_allele) {
    xa_mx[,i] <- xa_mx[,i]* -1
  }

}

xd_mx <- -2*abs(xa_mx)+1
# plot a histogram of the MAF values
xa_maf_calc <- xa_mx + 1
mafs <- apply(xa_maf_calc, 2, function(x) sum(x) / (length(x) * 2))
maf_hist <- hist(as.numeric(mafs), main = 'Distribution of Minor Allele Frequencies', xlab = 'Minor Alle
```



It looks good that most of the MAFs are < 0.5 which would be expected for an allele with the second lowest frequency. The cases where MAF > 0.5 I think means there is only one homozygous allele in the population and then a few heterozygotes and due to the method of calculation we get this value. I like to keep these out of the data to know that the variation is essentially fixed at this locus, so these data points would probably be best to throw out as not as useful, or the calculation could be adjusted.

d

The power of a hypothesis test is the probability of correctly rejecting the null hypothesis when it is incorrect, defined as $1 - \beta$ where β is the probability of committing a Type II error. That is to obtain a false negative, or incorrectly accept the null hypothesis.

This can be represented as $1 - \beta = \int_{c_\alpha}^{\infty} Pr(T(x)|\theta)dT(x)$

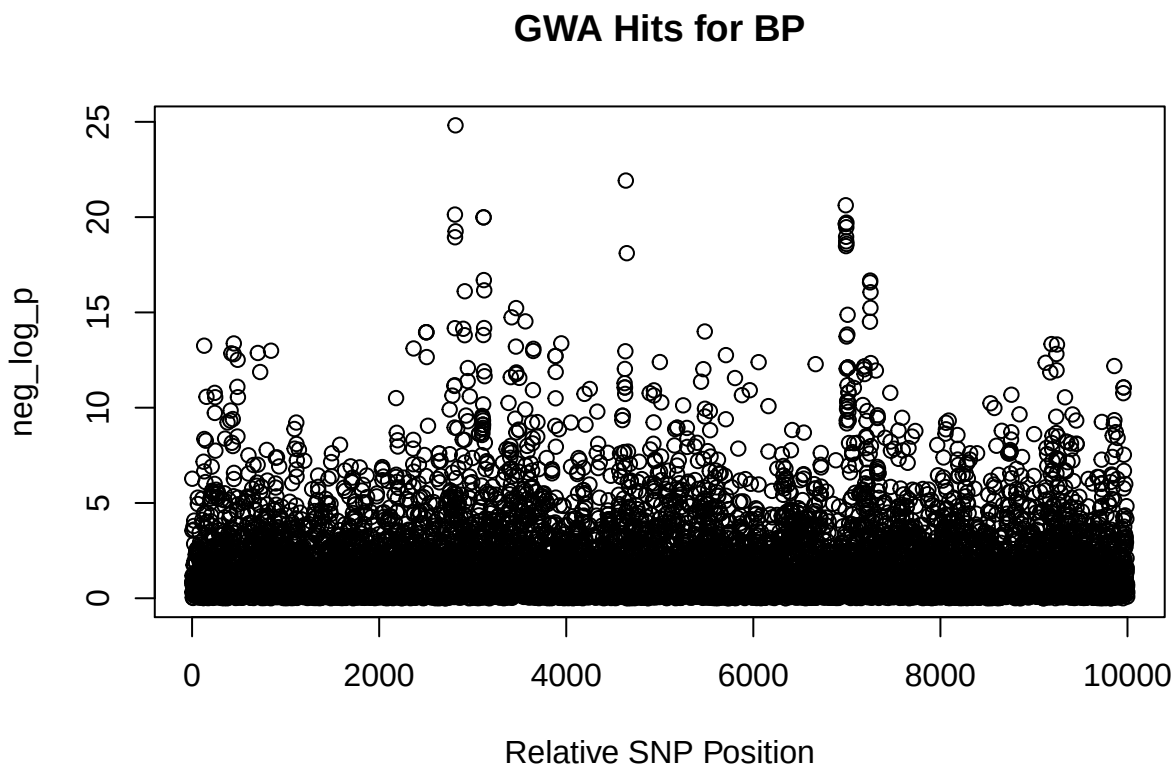
We cannot control the power directly. We can however, increase it by increasing the sample size, n, by decreasing the stringency of our selected α (increasing the value of α), choosing a one- or two-sided test based on the properties of the statistic we have calculated and with the design of the statistic used in the hypothesis test.

e

The MAF measures the frequency of the second most common allele in a population, therefore lower the MAF, the more homogenous the population is at a given loci. We are more capable of capturing the full variation of a potential phenotypic output from a given causal polymorphism when there is more variation at a given loci therefore ideally to have more power if we are directly measruing a causal polymorphism we would want a higher MAF.

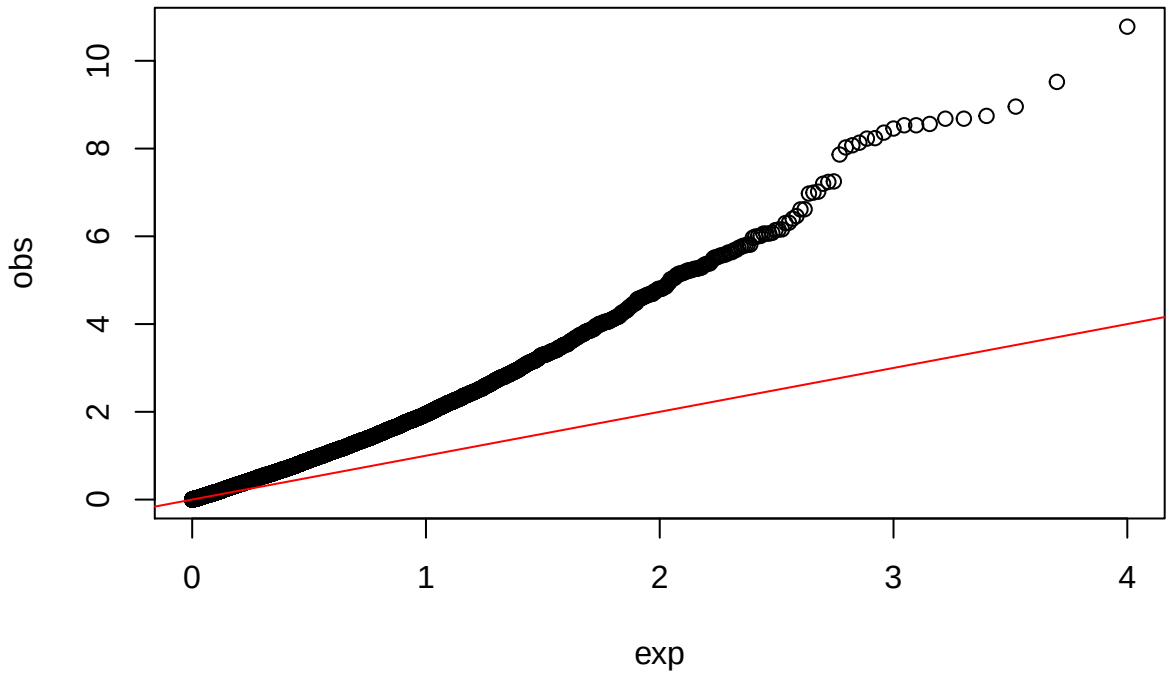
3

```r
# a MLE(betas) for a linear regression with no covariates, calculate p values for the f stat .. compari

# 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_mx)
pvals <- c()

# Calculate and save pvals for each phenotype-genotype pair

  for (i in 1 : n_geno){
      pvals <- c(pvals, pval_calculator(phenos[,2],
                                        xa_input = xa_mx[,i],
                                        xd_input = xd_mx[,i]))
  }

# b produce manhattan plot of the p values
neg_log_p <- -1 * log(pvals)

par(mfrow=c(1,1))
plot(neg_log_p, xlab = 'Relative SNP Position', main = 'GWA Hits for BP')
```



4

```r
# a produce a QQ plot for the p values in 3a
obs <- sort(-log10(pvals))
exp <- seq(from = (1 / length(obs)), to = 1, length.out = length(obs))
# - log 10
exp <- sort(-log10(exp))
plot(exp, obs)
abline(a=0,b=1,col='red')
```



b

No, it seems analyzing the data with this model would be inappropriate due to because the line curves upward. Ideally the QQ plot will be mostly a straight line because the observed p-values should mostly follow those that would result from a normally distributed sample of the given size under the null hypothesis, except where actually significant hits are found represented by a few jumps near the end of the scale. Bumps and waviness in the line like this signal there is an unreppresented structure in the data that is confounding our analysis of the relationships between genotypes and phenotypes

5

```
# a use substr(<string>, start, end) to calculate and report the number of people in each population (n
hg_count <- 0
na_count <- 0

for (i in 1:nrow(phenos)) {
  x <- phenos[i, 1]
  if(substr(x, 1, 2) == 'HG'){
    hg_count <- hg_count +1
  }
  else {
    na_count <- na_count + 1
  }
}
cat('There are n_1 = ', hg_count, ' people in the HG population and n_2 = ', na_count, ' people in the
```

```
## There are n_1 =  89  people in the HG population and n_2 =  92  people in the NA population.
```

b

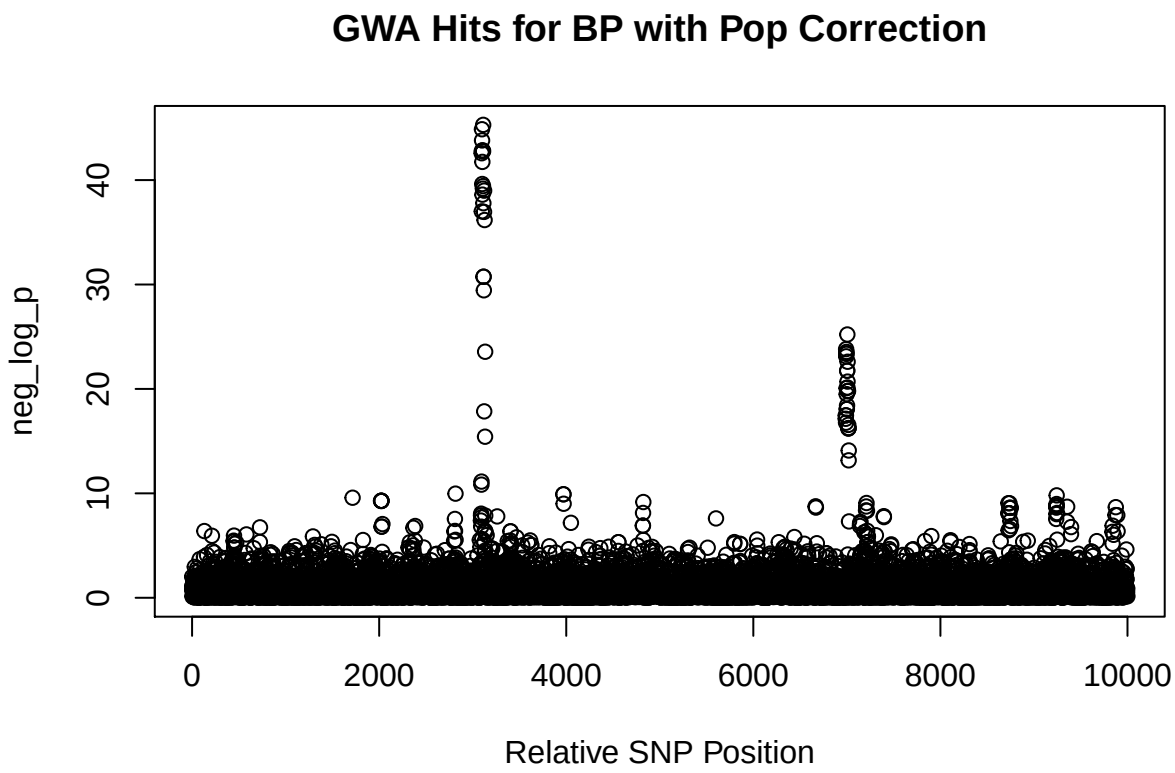
A PCA could have been used to ascertain the population structure of the data because a PCA calculates the eigen vectors and eigen values of a multi-dimensional dataset then reprojects the data set upon the given

number of reduced axes you choose. In this case, plotting the data along the first axis would likely have revealed two clusters representing the two separate populations because of the inherent relationships among the data along this new summary component for the data.

6

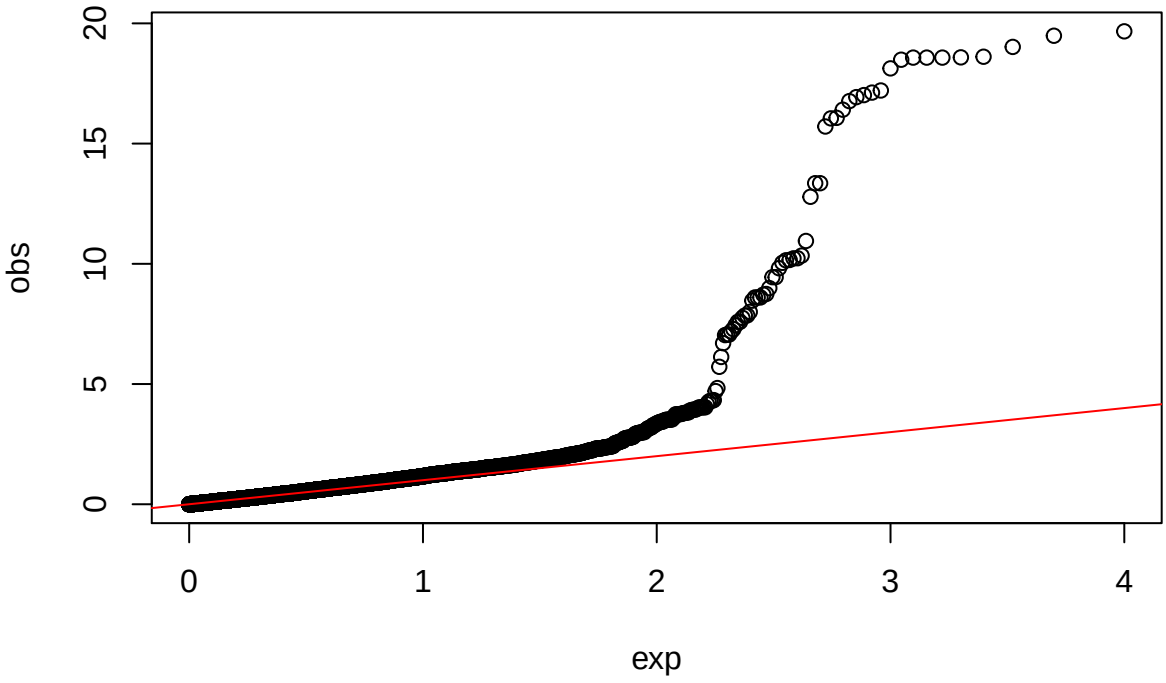
```r
# a calculate MLE(b) for null and alternative hypothesis with population structure covariate
# code populations as n_1 = -1, n_2 = 1
phenos$pop_encode <- ifelse(substr(phenos$V1, 1, 2) == 'HG', -1, 1)

# calculate f-stats by hand, then pf to get p values
# New function to calculate the pval given a set of individuals' phenotype, and genotype encodings, adj
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/ cova
    XZ_mx <- cbind(1,xa_input,xd_input,z_input)                                 # w/ genotype

    # 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/ cova
    MLE_beta_theta1 <- ginv(t(XZ_mx) %*% XZ_mx) %*% t(XZ_mx) %*% pheno_input    # w/ genotype

    # 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/ cova
    y_hat_theta1 <- XZ_mx %*% MLE_beta_theta1                                   # w/ genotype

    # 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/ cova
    SSE_theta1 <- sum((pheno_input - y_hat_theta1)^2)                           # w/ genotype

    # 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)
}
pvals <- matrix(data = NA, nrow = nrow(xa_mx))

for(i in 1:ncol(xa_mx)){
  pvals[i] <- pval_calculator_lab10(phenos[,2], xa_mx[,i], xd_mx[,i], phenos[,3])
}

# b make Manhattan plot of values
neg_log_p <- -1 * log(pvals)

par(mfrow=c(1,1))
```

```r
plot(neg_log_p, xlab = 'Relative SNP Position', main = 'GWA Hits for BP with Pop Correction')
```



7

```r
# a make QQ plot of p vals from 6a
obs <- sort(-log10(pvals))
exp <- seq(from = (1 / length(obs)), to = 1, length.out = length(obs))
# - log 10
exp <- sort(-log10(exp))
plot(exp, obs)
abline(a=0,b=1,col='red')
```



b

It appears that this correction makes inference from this data more likely to be accurate because the QQ plot now hugs the line much better for the relationship between the expected and observed small p values then jumps from this line where the p-values for potential causal polymorphisms in the dataset lie. The significant hits jump away from the expected p values if the null hypothesis were true because the current analysis found a relationship between these SNPs and a measurable difference in phenotype under the conditions of the experiment.

8

```
# a calculate the p value cut off for an alpha of 0.05 w Bonferroni correction
cat('The p-value cutoff for a Bonferroni multiple test correct with alpha = 0.05, would be ', 0.05/n_ge
```

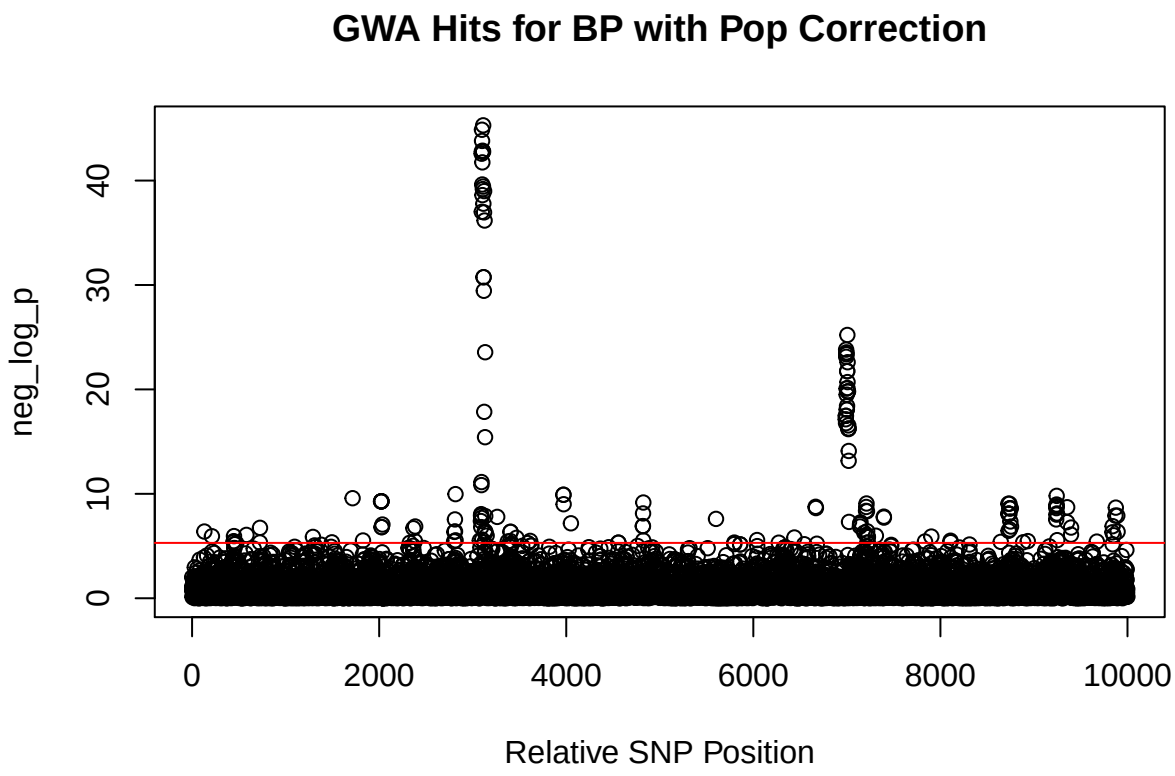
```
## The p-value cutoff for a Bonferroni multiple test correct with alpha = 0.05, would be  5.001e-06 . T
```

b (2 sent)

A Manhattan plot with the new stringency level indicated by a red line is shown below.

From this, I would estimate about 10 significant areas of the genome should be searched first in association with the BP reading phenotype. I determined this by only considering distinct, tall peaks that seem to rising out of the SNP cloud at the bottom, above the red line and that consist of dots all along the scale of p-vals (not a spurious sequencing error etc.) and those that are far enough away from each other (a clear gap since we don't have the actual genomic bp positions of the SNPs) to not be in linkage disequilibrium and therefore representing unique areas.

```
plot(neg_log_p, xlab = 'Relative SNP Position', main = 'GWA Hits for BP with Pop Correction')
abline(h=-log10(0.05/n_geno),col='red')
```



c

It's likely only one causal polymorphism is indicated by each peak. Molecular biology dictates that sections of chromosomes that are close together are more likely to be inherited together, therefore SNPs that are near each other (and therefore close enough to the causal polymorphism) will exhibit a similar GWA signal.

d

We have not measured every basepair of the genome, so we cannot say which specific polymorphism is specifically associated with a given phenotype, just the area of the genome in which this specific variation occurs. We can do this mapping however because of the convenient fact that areas of the genome that are close together are more likely to be inherited together.

9

```
# a list the p value and number (1...N) of the most significant SNP for each peak
# obtain max p val for in an area near the peaks I deem significant
sig_areas <- data.frame('starts' = c(1900, 2600, 2900, 3800, 4700, 6700, 6900, 8700, 9300, 9800), 'ends


# calculate the max p value in these ranges and their corresponding locations in the SNP data
highest_ps <- matrix(data = NA, nrow = length(sig_areas[,1]))
snp_positions <- matrix(data = NA, nrow = length(sig_areas[,1]))
for (i in 1:length(highest_ps)) {
  highest_ps[i] <- max(pvals[sig_areas$starts[i]: sig_areas$ends[i]], na.rm = TRUE)
```

```
  snp_positions[i] <- sig_areas$starts[i] + which.max(pvals[sig_areas$starts[i]: sig_areas$ends[i]]) - 1
}

cat('The maximum p values obtained for each peak are ', highest_ps, ' at positions ', snp_positions, ',
```

```
## The maximum p values obtained for each peak are  0.9999886 0.9953017 0.9970436 0.9996413 0.9391915 0
```

b

The relationship that governs how segments of chromosomes are experience crossing over in meiosis is largely uncoupled from the heritability of traits therefore it's possible that SNPs closer to the causal polymorphism experience less linkage with this trait than those that are further away. This is why it is important to consider entire windows of linkage disequilibrium when significant SNPs are found to make sure that we are considering all possible causal polymorphisms.

```
# c for the most significant SNP in each peak, calculate its correlation with the closest SNPs on eithe

expect <- function(x){
  tbl <- table(x)
  ex <- 0
  for(i in 1:length(tbl)){
    # (x = i) * pr(x = i)
    ex <- ex + as.numeric(names(tbl[i])) * (tbl[i] / length(x))
  }
  return(ex)
}

vari <- function(x){
  tbl <- table(x)
  vernc <- 0
  for (i in 1:length(tbl)) {
    # (x = i) - ex^2 * pr(x = i)
    vernc <- vernc + ((as.numeric(names(tbl[i])) - expect(x))^2 * (tbl[i] / length(x)))
  }
  return(vernc)
}

covar <- function(x1, x2){
  return(expect((x1 - expect(x1))*(x2 - expect(x2))))
}

corlat <- function(snp, partner, xas){
  snp1 <- xas[,snp]
  snp2 <- xas[,partner]
  kerr <- covar(snp1, snp2) / (sqrt(vari(snp1)) * sqrt(vari(snp2)))
  return(kerr)
}

corr_rights <- matrix(data = NA, nrow = length(highest_ps))
corr_lefts <- matrix(data = NA, nrow = length(highest_ps))

for (i in 1:length(highest_ps)) {
  corr_rights[i] <- corlat(snp_positions[i], snp_positions[i] + 1, xa_mx)
```

```
  corr_lefts[i] <- corlat(snp_positions[i], snp_positions[i] - 1, xa_mx)
  #corr_rights[i] <- cor(xa_mx[,snp_positions[i]], xa_mx[,snp_positions[i] + 1])
}

result <- data.frame('snp' = snp_positions, 'right_corr' = corr_rights, 'left_corr' = corr_lefts)

cat('The SNP positions with the highest p values and their correlations with right and left nearest SNPs
```

```
## The SNP positions with the highest p values and their correlations with right and left nearest SNPs a
```

```
print.data.frame(result)
```

```
##     snp  right_corr   left_corr
## 1  2043  0.98611652  0.97967481
## 2  2776  0.11922902 -0.10838144
## 3  3074 -0.53904551  0.99208265
## 4  3803  0.14701590 -0.03011404
## 5  4737  0.22804761 -0.54968968
## 6  6717  0.02691328  0.15331788
## 7  6904 -0.33039747  0.60179190
## 8  8762  0.03768051 -0.12931357
## 9  9533 -0.61006970 -0.32210107
## 10 9806  0.49208058  0.42691762
```

d

A correlation of zero indicates that the random variables compared (in this case a SNP vector with then vector of either of its two neighbors) are independent. We know that segments of chromosomes that are near each other tend to be inherited together ("linked" in linkage disequilibrium) and thus are predictive of each other's state and should not appear independent (i.e. $Corr(X_1, X_2) \neq 0$).

10

a

A causal polymorphism is a position in the genome where an experimental manipulation of the DNA produces an effect on the phenotype on average or under specified conditions. This can be represented by $A_1 \rightarrow A_2 \implies \Delta Y|Z$

b

In an ideal experiment, we would need to have sampled every member of the population at every location in their chromosome in order to rigorously assert that a specific polymorphism is causal. This way our model would fully capture the variation in the population and we would no longer be conducting inference on the frequencies of alleles and their relationships to phenotypes in the population, we would just be directly calculating the relationships.

c

A p-value is the probability of obtaining a value of a statistic $T(x)$ or more extreme, conditional on the null hypothesis, H_0 being true. It is a function of a statistic defined as $pval = Pr(|T(x)| \geq t|H_0 : \theta = c)$.

d

There are many things that can cause problems in GWAS that lead to false positives. These include sequencing errors – in which a SNP is miscalled and consequently leads to phantom peaks, situations where disequilibrium occurs but there is no linkage – for example chromatin structure and other factors that cause far away regions of chromosomes to be inherited together, and unaccounted for covariates like population structure in which different allele fequencies confound the interpretation of the impact of X_a and X_d variables on phenotypes when not factored in to the regression. Some of these factors are easier to identify and address than others.